2009-09

Identification and classification of OFDM based signals using preamble correlation and cyclostationary feature extraction

Schnur, Steven R.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IDENTIFICATION AND CLASSIFICATION OF OFDM BASED SIGNALS USING PREAMBLE CORRELATION AND CYCLOSTATIONARY FEATURE EXTRACTION

by

Steven R. Schnur

September 2009

Thesis Co-Advisors: Murali Tummala
John McEachen

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Identification and Classification of OFDM Based Signals using Preamble Correlation and Cyclostationary Feature Extraction		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Steven R. Schnur			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE** A	

13. ABSTRACT (maximum 200 words)

In this thesis, a scheme for the identification and classification of orthogonal frequency division multiplexing based signals is proposed. Specifically, the cyclostationary signature of IEEE 802.11 and IEEE 802.16 standard compliant waveforms is investigated. A model is introduced that identifies the waveform; in the case of IEEE 802.11, confirms identification decision via cyclostationary feature extraction. If the waveform is identified as being IEEE 802.16 compliant, the scheme will classify the cyclic prefix size of the waveform. After cyclic prefix classification, the 802.16 waveform will be subjected to cyclostationary feature extraction for identification confirmation. The cyclostationary signature of each waveform is generated via a computationally efficient algorithm called the fast Fourier transform accumulation method, which produces an estimate of the waveform's spectral correlation density function. Simulation results based on MATLAB implementation are presented.

14. SUBJECT TERMS IEEE 802.11, IEEE 802.16, OFDM, Cyclostationary Feature Extraction, FFT Accumulation Method			**15. NUMBER OF PAGES** 123
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

IDENTIFICATION AND CLASSIFICATION OF OFDM BASED SIGNALS USING PREAMBLE CORRELATION AND CYCLOSTATIONARY FEATURE EXTRACTION

Steven R. Schnur
Major, United States Marine Corps
B.S., Pennsylvania State University, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: Steven Schnur

Approved by: Professor Murali Tummala
Thesis Co-Advisor

Professor John McEachen
Thesis Co-Advisor

Professor Jeffery B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, a scheme for the identification and classification of orthogonal frequency division multiplexing based signals is proposed. Specifically, the cyclostationary signature of IEEE 802.11 and IEEE 802.16 standard compliant waveforms is investigated. A model is introduced that identifies the waveform and, in the case of IEEE 802.11, confirms identification decision via cyclostationary feature extraction. If the waveform is identified as being IEEE 802.16 compliant, the scheme will classify the cyclic prefix size of the waveform. After cyclic prefix classification, the 802.16 waveform is subjected to cyclostationary feature extraction for identification confirmation. The cyclostationary signature of each waveform is generated via a computationally efficient algorithm called the fast Fourier transform accumulation method, which produces an estimate of the waveform's spectral correlation density function. Simulation results based on MATLAB implementation are presented.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
- **A. BACKGROUND 1**
- **B. OBJECTIVE AND APPROACH 2**
- **C. RELATED WORK 2**
- **D. ORGANIZATION OF CHAPTERS 3**

II. REVIEW OF OFDM AND CYCLOSTATIONARITY 5
- **A. OFDM PHYSICAL LAYER FUNDAMENTALS 5**
- **B. IEEE 802.11-2007 SPECIFICATIONS 10**
  - **1. Physical Layer 10**
  - **2. IEEE 802.11 Medium Access 14**
- **C. IEEE 802.16-2004 SPECIFICATIONS 14**
  - **1. IEEE 802.16 Physical Layer 15**
  - **2. IEEE 802.16 Medium Access 19**
- **D. CYCLOSTATIONARITY 19**

III. OFDM BASED SIGNAL IDENTIFICATION AND CLASSIFICATION 25
- **A. WAVEFORM IDENTIFICATION AND CLASSIFICATION 25**
- **B. WAVEFORM IDENTIFICATION BY PREAMBLE CROSS-CORRELATION 27**
- **C. CYCLOSTATIONARY FEATURE EXTRACTION 31**
  - **1. Frame Preamble Cyclostationary Signature 31**
    - ***a. IEEE 802.11 Preamble Considerations 31***
    - ***b. IEEE 802.16 Preamble Considerations 31***
  - **2. Pilot Subcarrier Cyclostationary Signature 32**
  - **3. Embedded Cyclostationary Signature 34**
- **D. FFT ACCUMULATION METHOD (FAM) 36**

IV. BASEBAND SIMULATION RESULTS 41
- **A. SIMULATION MODEL 41**
- **B. IMPLEMENTATION 42**
  - **1. Baseband Signal Generation 43**
  - **2. Preamble Cross-correlation Identification 45**
  - **3. 802.16 CP Classification 46**
  - **4. Cyclostationary Waveform Identification 50**
    - ***a. IEEE 802.11 Cyclostationary Feature Extraction 50***
    - ***b. IEEE 802.16 Cyclostationary Feature Extraction 51***
  - **5. Baseband receiver 52**
- **C. RESULTS 54**
  - **1. Preamble Cross-Correlation 54**
  - **2. IEEE 802.16 CP Classification 57**
  - **3. Cyclostationary Feature Extraction 58**
    - ***a. IEEE 802.11 ..... 58***
    - ***b. IEEE 802.16 ..... 63***

V. CONCLUSIONS 69
A. SIGNIFICANT RESULTS AND CONTRIBUTIONS 69
B. FUTURE WORK 70
APPENDIX 71
LIST OF REFERENCES 103
INITIAL DISTRIBUTION LIST 105

LIST OF FIGURES

Figure 1. Basic Multicarrier Modulation (After [5]). 6
Figure 2. Orthogonal Subcarrier Spacing through IFFT Implementation. 6
Figure 3. OFDM Block Description for 16 point IFFT. 8
Figure 4. OFDM Frequency Subcarrier Description of a 16 Point IFFT System. 8
Figure 5. 1/4 Length CP Where the Last $N_{FFT}/4$ Samples are Copied and Appended to the Leading Edge of the Symbol. 9
Figure 6. PPDU Frame Format (From Ref [8]). 10
Figure 7. OFDM training Structure for 20 MHz Channel Spacing Option (After [8]). 12
Figure 8. TDD Frame Structure (After [7]). 16
Figure 9. Downlink and Network Entry Preamble Structure (After [6]). 17
Figure 10. PRBS generator for IEEE 802.16 Pilot Subcarrier Modulation (After [7]). 18
Figure 11. Proposed Waveform Identification and Classification Model. 26
Figure 12. Cross-Correlation of an IEEE 802.11 Preamble Versus (a) 802.11 Sample Preamble, (b) 802.16 Sample Preamble. 29
Figure 13. Cross-Correlation of 802.11 Preamble Versus a) 802.11 Sample Preamble, b) 802.16 Sample Preamble. 30
Figure 14. Spectral Line Development: (a) 802.16 Waveform SCD with Constant Pilot Subcarrier Sequence. (b) 802.16 Waveform PSD with Constant Pilot Subcarrier Sequence. 33
Figure 15. An Example of Subcarrier Set Mapping to Establish an Embedded Cyclostationary Signature. 35
Figure 16. Block Diagram FFT Accumulation Method. (From [11]). 38
Figure 17. Region of support within the bifrequency plane (From [2]). 39
Figure 18. MATLAB Implementation Model. 42
Figure 19. Flow Chart of Main Program. 43
Figure 20. Generic flow of Signal Generator. 44
Figure 21. I-Q Voltage Constellation of Quadrature Amplitude Modulation 64 with DC/Guard Nulls and Pilot Symbol Locations. 45
Figure 22. Program Flow to Classify IEEE 802.16 CP length. 47
Figure 23. Matrix Description of FAM SCD function estimator Output, $S_{xx}^{\alpha}(f)$ 49
Figure 24. Received I-Q Voltage Constellation of Quadrature Amplitude Modulation 64 with Channel Conditions Simulating a SNR of 20 dB. 54
Figure 25. Preamble Correlation Results of an IEEE 802.11 Received Waveform versus SNR Averaged Over 150 Runs. 55
Figure 26. Preamble Correlation Results of an IEEE 802.16 Received Waveform versus SNR Averaged Over 150 Runs. 56
Figure 27. CP Classification Percentage versus the Number of OFDM Symbols Processed by the Test FAM SCD Function Estimator. Averaged Over 200 Runs. 58
Figure 28. Surface Plot Representation of an IEEE 802.11 Waveform's $S_{xx}^{\alpha}(f)$, Averaged Over 120 OFDM Symbols. 59

Figure 29. Profile Plots: Magnitude of $S_{xx}^{\alpha}(f)$ versus a) Cyclic Principal Frequency and b) Frequency for IEEE 802.11 Waveform. Results Averaged Over 120 OFDM Symbols. 60
Figure 30. Contour Plot of an IEEE 802.11 Waveform's $S_{xx}^{\alpha}(f)$ Magnitude. . Results Averaged Over 120 OFDM Symbols. 61
Figure 31. IEEE 802.11 Subcarrier SCD Values versus SNR. 62
Figure 32. IEEE 802.11 Averaged Subcarrier SCD Values versus Number of OFDM Symbols Processed by the FAM SCD Function Estimator. 62
Figure 33. Surface Plot Representation of an IEEE 802.16 Waveform's $S_{xx}^{\alpha}(f)$. Results Averaged Over 120 Symbols. 63
Figure 34. Profile Plots: Magnitude of $S_{xx}^{\alpha}(f)$ versus a) Cyclic Principal Frequency and b) Frequency for IEEE 802.16 Waveform. Results Averaged Over 120 Symbols....64
Figure 35. Contour Plot of an IEEE 802.16 Waveform's $S_{xx}^{\alpha}(f)$ Magnitude. Results Averaged Over 120 Symbols. 65
Figure 36. IEEE 802.16 Averaged Subcarrier Peak Values versus SNR. 66
Figure 37. IEEE 802.16 Averaged Subcarrier Peak Values versus Number of OFDM Symbols Processed by the FAM SCD Function Estimator. 66

LIST OF TABLES

Table 1: IEEE 802.11 Major OFDM PHY Parameters (From [8]).....13

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Developing countries around the world are providing broadband wireless access to significant portions of their populations through Wireless Local Area Network (WLAN) and Wireless Metropolitan Area Network (WMAN) technologies. Many of these developing countries currently implementing broadband wireless networks are located in regions that possess potential or known threats to the security of the United States of America. This has lead to an increased interest in the ability to identify and classify these types of signals.

There are three well-known methods of signal identification and classification currently being investigated for feasibility in this endeavor: energy detection, matched filter detection and cyclostationary feature detection. This thesis employs the cyclostationary feature method, due to the cyclostationary characteristics present in Orthogonal Frequency Division Multiplexing (OFDM) modulated signals. OFDM is the modulation scheme utilized in the IEEE 802.11-2007 WLAN and 802.16-2004 WMAN standards because of its resistance to multipath fading.

The objective of this thesis is to develop a method of identifying and classifying OFDM based wireless data and communication network waveforms. In order to accomplish this, a scheme is developed to integrate the processes of initial waveform identification, cyclic prefix classification and cyclostationary feature extraction. Waveform identification is accomplished through a preamble correlation process. The results of the preamble correlation process determine if a received waveform is IEEE 802.11 or 802.16 standard compliant.

Cyclostationary feature extraction is employed to classify the cyclic prefix of IEEE 802.16 waveforms, as well as confirm the identification results of preamble correlation for both waveforms. The method of analyzing the cyclic spectral properties of OFDM signals is implemented through the fast Fourier transform Accumulation Method (FAM). The FAM algorithm produces an estimate of the analyzed signal's cyclic spectral features. Additionally, the FAM provides a method of examining both the

power spectral density and the cyclic spectral density commonly referred to as the spectral correlation density function. All results were generated through MATLAB simulation.

There are four major contributions made by this thesis. First, the work established an approach to identify a received waveform with no a priori knowledge of its origin. After much research and experimentation, it was determined that a preamble cross-correlation operation provides satisfactory results while minimizing computational complexity of the simulations.

Next, a method of classifying the cyclic prefix length of IEEE 802.16 waveforms is devised. Since the cyclic prefix option is established when the network is first setup, there is no way for a passive listener to identify this value by decoding captured control transmissions. By employing cyclostationary feature extraction, we are able to determine the cyclic prefix length in a rapid fashion.

After identifying three methods of implementing cyclostationary feature extraction from [1], [3] and [4], we needed to determine which method or methods would be most compatible with the stated objectives of this work. Although embedded cyclostationary signature extraction is a promising technique, pilot subcarrier cyclostationary feature extraction proved to be the most applicable method.

ACKNOWLEDGMENTS

This endeavor would not have been possible without the support and understanding of my wife, Eileen, and son, Owen.

Thank you, Dr. Tummala, for the outstanding guidance and insight you provided throughout the thesis process.

Dr. McEachen, thank you for your assistance in finalizing this work.

Dr. Ha, thank you for the outstanding instruction and guidance you provided.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The rapidly advancing technologies of wireless communication networks are providing enormous opportunities. A large number of users in emerging markets throughout the world are gaining access to these networks. Developed countries possess extensive wired communications networks, but developing countries do not have the financial resources to implement the infrastructure necessary to provide access for significant portions of their populations. With the implementation of relatively affordable technologies based on the Institute of Electrical and Electronics Engineers (IEEE) Wireless Local Area Network (WLAN) and Wireless Metropolitan Area Network (WMAN) standards, these developing countries now have the ability to provide that access. Many of the developing countries currently implementing these broadband wireless networks are located in regions that possess potential or known threats to the security of the United States of America. This has lead to an increasing amount of interest in the ability to identify and classify these types of signals.

There are three well-known methods of signal identification and classification currently being investigated for feasibility in this endeavor: energy detection, matched filter detection, and cyclostationary feature detection. This thesis employs the cyclostationary feature method, due to the cyclostationary characteristics present in Orthogonal Frequency Division Multiplexing (OFDM) modulated signals [1]. OFDM is utilized in the IEEE 802.11-2007 and 802.16-2004 standards. The communications industries refer to the implementation of these standards as Wi-Fi (IEEE 802.11) and Worldwide interoperability for Microwave Access (IEEE 802.16), respectively. Any further mention of IEEE 802.11 and IEEE 802.16 in this thesis will be in reference to the IEEE 802.11 and 802.16 standards, respectively.

Traditional signal analysis methods typically employ stationary probabilistic models that examine a small time sample of a signal and assume that the statistical properties of the signals are time invariant. The term that identifies this stationary

property with respect to first and second order statistical moments is Wide Sense Stationarity (WSS). The WSS property, which is satisfied if the signal's mean is constant and autocorrelation is independent of time, ignores the statistically periodic or "cyclic" properties of manmade signals that can be developed. These cyclostationary properties can be very useful in distinguishing a variety of random signals from one another.

B. OBJECTIVE AND APPROACH

The objective of this thesis is to develop a method of identifying and classifying OFDM based wireless data and communication network waveforms. Waveform identification will be accomplished through a preamble correlation process. The results of the preamble correlation process will determine if a received waveform is IEEE 802.11 or 802.16 standard compliant.

Cyclostationary feature detection will be employed to classify the Cyclic Prefix (CP) of IEEE 802.16 waveforms and then confirm the identification results of preamble correlation. The method of analyzing the cyclic spectral properties of OFDM signals will be implemented through the Fast Fourier Transform (FFT) Accumulation Method (FAM). The FAM algorithm produces an estimate of the analyzed signal's cyclic spectral features through MATLAB simulation. Additionally, the FAM provides a method of examining both the Power Spectral Density (PSD) and the cyclic spectral density commonly referred to as the Spectral Correlation Density (SCD) function. All results will be generated through MATLAB simulation.

C. RELATED WORK

Most of the background information that exists on cyclostationarity has been influenced by W. A. Gardner's work. In particular, [2] is a comprehensive compilation of works on cyclostationarity.

Although there is a significant amount of literature on cyclostationarity, there are only a few articles that extend this method to OFDM based signals. A significant amount of insight into properties of OFDM that might exhibit recognizable cyclostationary signatures are provided by [1], [3], and [4].. In particular, [1] focuses on a

cyclostationary signature produced by pilot subcarriers which are embedded within each OFDM symbol. Unlike the approach utilized in [1], which analyzed waveforms that applied a 2.5 dB gain to the pilot subcarriers, the waveforms investigated in this work did not amplify the pilot subcarriers instead an averaging process was implemented. We will investigate two additional cyclostationary signature approaches mentioned [3] and [4], although their application was deemed incompatible with the stated objectives of this thesis.

D. ORGANIZATION OF CHAPTERS

Chapter II provides a basic review of OFDM and cyclostationarity. The review of OFDM is by no means comprehensive and mainly focuses on the properties that are exploited for identification and classification purposes. The chapter will examine why OFDM modulation is becoming so popular and how it is utilized within the IEEE 802.11 and 802.16 standards. Concluding the chapter will be a discussion on cyclostationarity.

Chapter III introduces a method by which IEEE 802.11 and 802.16 waveforms will be identified and classified. A model is proposed that will identify and classify IEEE 802.11 and 802.16 standard compliant waveforms. Included in this discussion is a brief look at the FAM estimator that is employed to generate the SCD function of OFDM based waveforms. Chapter IV introduces the MATLAB implementation of the model described in Chapter III. An overview of the program utilized to generate, identify and classify IEEE 802.11 and 802.16 compliant waveforms is performed. Included is a discussion of how the FAM is modified and improved to facilitate identification and classification of these waveforms. Results of MATLAB simulation of the proposed identification and classification scheme are presented.

Chapter V will provide a brief summary of work performed and discuss significant results obtained and opportunities for future work related to identification and classification of broadband wireless networking waveforms utilizing cyclostationary techniques.

The Appendix will consist of MATLAB code that was developed for simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

II. REVIEW OF OFDM AND CYCLOSTATIONARITY

The most common metric utilized to measure the performance of communication networks is transmission bit rate. As the shear volume of data that is required to be transmitted over wireless channels consistently increases due to the development of more complex applications, the demand for higher transmission bit rates appears to be insatiable. Unfortunately, in a wireless channel, as the transmission bit rate increases, a phenomenon known as multipath fading increases as well, thus driving bit transmission errors toward unacceptable levels. The most effective method in combating multi-path fading at high bit transmission rates is multi-carrier modulation, more specifically OFDM. In the following sections, we will take a detailed look at the fundamentals of OFDM and how it is implemented by the IEEE 802.11 and 802.16 standards. Finally, the concept of cyclostationarity will be introduced.

A. OFDM PHYSICAL LAYER FUNDAMENTALS

The concept of multi-carrier modulation has existed since the early 1960s. It is a simple yet effective solution to the multipath fading imposed bit rate limitation of the channel. The effects of multi-path fading are negligible at relatively low bit rates. At higher bit rates, the period of a data symbol is reduced and becomes comparable to the duration of the delay spread in the channel leading to Inter-Symbol Interference (ISI).

In order to overcome ISI, multi-carrier modulation breaks up the high bit rate serial stream that is to be transmitted into lower rate parallel bit streams. This is accomplished by dividing the serial bit stream of rate R into L parallel lower rate bit streams, of R/L, which are then modulated onto L subcarriers distributed within a specified frequency band. Once the L bit streams are modulated to their respective subcarriers, the modulated subcarriers are summed and transmitted. Figure 1 depicts the concept of multicarrier modulation.

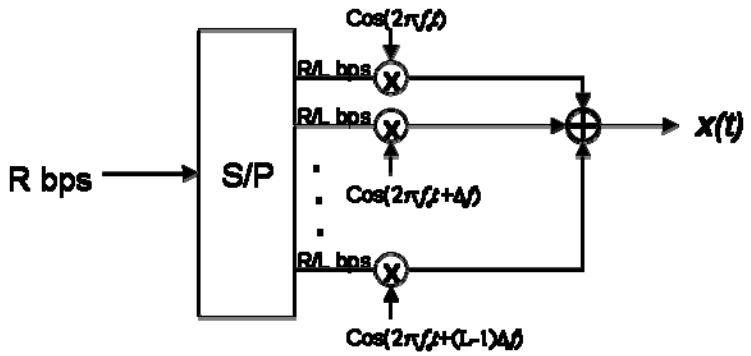

Figure 1. Basic Multicarrier Modulation (After [5]).

Effectively, the original bandwidth *W* of the high rate bit stream *R* is segmented into *L* subbands of bandwidth *W*/*L*. The subcarriers are separated by an optimal distance within the frequency band, referred to as orthogonality, to avoid inter carrier interference. This is accomplished through the effective use of digital signal processing techniques such as FFT and Inverse FFT (IFFT) in the baseband, prior to RF modulation. To illustrate the orthogonal subcarrier separation, Figure 2 shows the frequency response of adjacent subbands in a FFT. To visualize orthogonality, notice that when the response of any one subband is at its maximum, the collection of spurious responses from all the remaining subbands is zero.

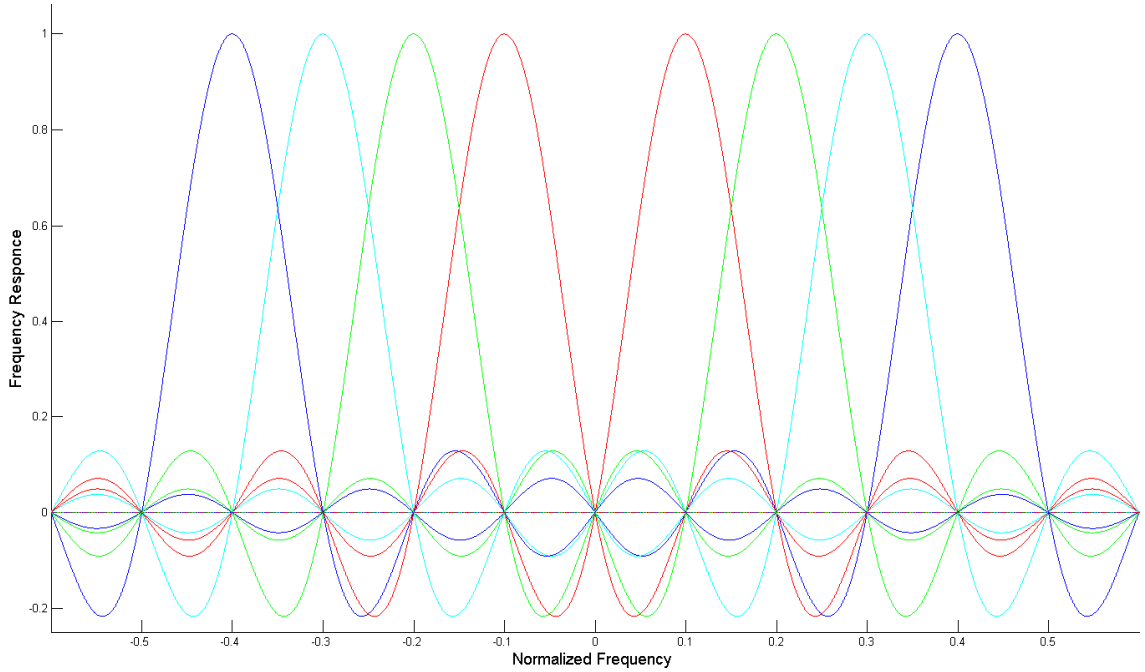

Figure 2. Orthogonal Subcarrier Spacing through IFFT Implementation.

The process of OFDM modulation is depicted in Figure 3. Initially, the high rate serial bit stream R is converted into L parallel lower rate bit streams. Next each bit stream is segmented into groups of bits called symbols. The size of the grouping is dependent on channel conditions; the better the channel state the larger the number of bits used to form the symbol and vice-versa. The purpose of the grouping is to prepare the bits for baseband modulation or symbol mapping. The number of symbol levels M is defined as

$$M = 2^k \ , \tag{2.1}$$

where k is the number of bits per symbol. The symbol period is defined as

$$T = kT_b \, , \tag{2.2}$$

where T_b is the bit period. The baseband modulation methods employed in most OFDM systems consist of Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), Quadrature Amplitude Modulation 16 (QAM-16), and Quadrature Amplitude Modulation 64 (QAM-64) for $k = 1, 2, 4, 6$, respectively. The output of the baseband modulation is data symbol X_k, a complex valued term with in-phase and quadrature-phase (I-Q) components. Next the L data symbols are input to an IFFT operation, given as follows [6],

$$x[n] = \sum_{k=-N_{IFFT}/2}^{N_{IFFT}/2-1} X_k e^{j2\pi kn\Delta f} \tag{2.3}$$

where N_{IFFT} is the size of the IFFT and Δf is the subcarrier spacing in Hz.

Typically, not all of the IFFT inputs are utilized as data subcarriers. A certain number of subcarriers are used as pilot subcarriers for channel estimation. Additionally, numerous lower and upper end subcarriers are set to zero to reduce adjacent channel interference. The number of pilot and null or guard subcarriers depend on the size of the IFFT and the standard governing the communication system. The pilot subcarriers transmit a pseudo-random sequence that is known by the receiver. This allows for determination of channel conditions and therefore which baseband modulation technique to employ. The pseudo-random pilot sequence will be an important feature for system identification as will be demonstrated later in this thesis.

The inputs to the IFFT that are not nulls or pilot subcarriers are utilized as data subcarriers. For instance, the system in Figure 3 illustrates the example of a 16 point IFFT with ten data subcarriers, two pilot subcarriers and three guard subcarriers. The guard subcarriers are located in the two most negative frequency indexed subcarrier positions and the most positive frequency indexed subcarrier location. The 0^{th} subcarrier null is referred to as the "DC" null and serves as the center point or zero frequency subcarrier.

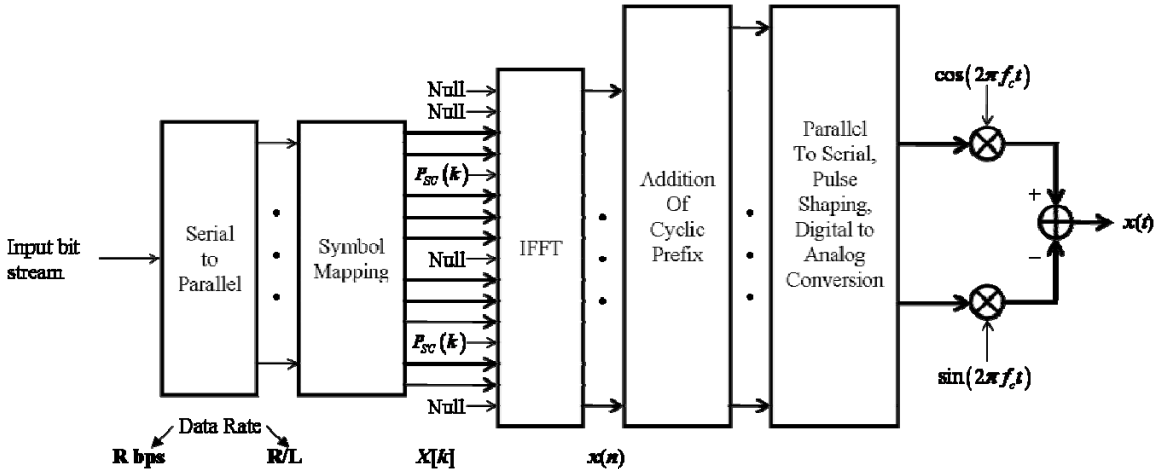

Figure 3. OFDM Block Description for 16 point IFFT.

Figure 4 depicts the spectral layout of the guard, pilot and data subcarriers of the OFDM transmitter depicted in Figure 3. Additionally, Figure 4 depicts the subcarrier frequency offset index, which is a reference for where the subcarrier is positioned within the waveforms spectral bandwidth.

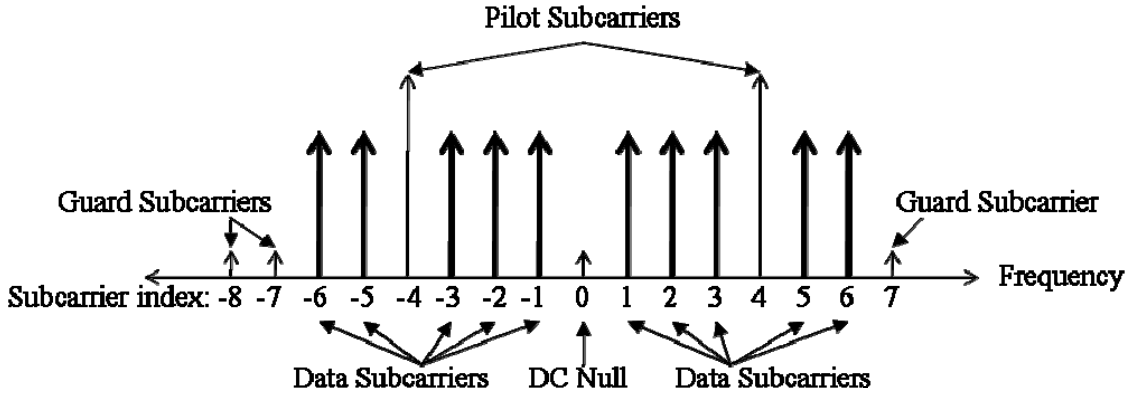

Figure 4. OFDM Frequency Subcarrier Description of a 16 Point IFFT System.

Next, a Cyclic Prefix (CP) or guard interval is inserted into the OFDM symbol that was formed from the output samples of the IFFT. The CP is a portion of the higher index IFFT output samples. These samples are copied and appended to the OFDM symbol as the leading portion. The purpose of the CP is to prevent ISI among OFDM symbols during transmission through a multipath fading channel; CP size is dependent on channel conditions and the governing standard.

Referencing the IEEE 802.16 standard [7], CP options consist of 1/4, 1/8, 1/16, and 1/32 of the IFFT size. If channel fading conditions are severe, the longest CP size will be selected which would be 1/4 length of the 256 point IFFT or 64 samples long. Figure 5 illustrates how a 1/4 length CP on a 256 sample OFDM symbol is created. This is accomplished by taking the last 64 samples and appending them to the front of the OFDM symbol. Unfortunately, there is a price to be paid for the addition of the CP onto each OFDM symbol. On the receiver end of the transmission, the CP is discarded. Because the CP is redundant information that is discarded, it is pure overhead which adds no useful data to the transmission and reduces throughput.

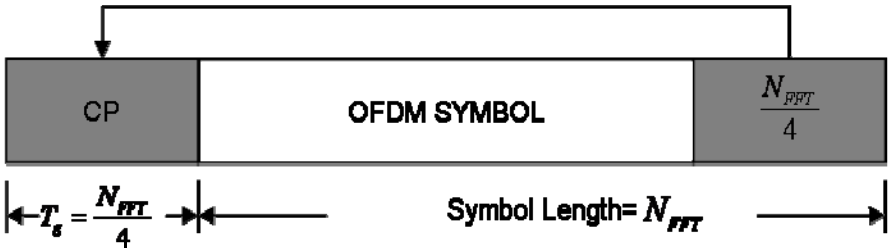

Figure 5. 1/4 Length CP Where the Last N_{FFT} / 4 Samples are Copied and Appended to the Leading Edge of the Symbol.

Now that the OFDM symbols have been formed and are ordered in a serial fashion, the samples are converted into a continuous-time signal by passing them through a digital to analog conversion and pulse shaping operation. The continuous time signal is then up converted to a radio frequency carrier and amplified for transmission.

Since the fundamentals of the OFDM Physical Layer (PHY) that pertain to this thesis have been discussed, we will now take a very brief look at the IEEE 802.11 WLAN

and 802.16 WMAN standards. The purpose of this discussion will be to focus on the portions of these two standards that will play a key role in the identification and classification of the respective OFDM signals.

B. IEEE 802.11-2007 SPECIFICATIONS

We have developed a basic understanding of OFDM in the previous section. The discussion will now focus on how it is employed within the IEEE 802.11 standard [8].

1. Physical Layer

For the discussion of the IEEE 802.11 standard, we will begin with the physical layer (PHY). The IEEE WLAN standard divides its PHY into two functional sublayers: The Physical Layer Convergence Procedure (PLCP) and the Physical Medium Dependent (PMD). The PLCP allows the Medium Access Control (MAC) layer to operate with minimum reliance on the PMD. The PMD essentially performs the role of actually transmitting and receiving data. The PLCP organizes MAC Protocol Data Units (MPDU) in to a desired frame format. These frames are then handed off to the PMD for transmission.

Figure 6 illustrates the structure of the PLCP frame. The PLCP preamble and the data OFDM symbols are the key points of interest of Figure 6, with respect to this thesis. These two items form the basis of an identification and classification scheme that will be introduced in Chapter III.

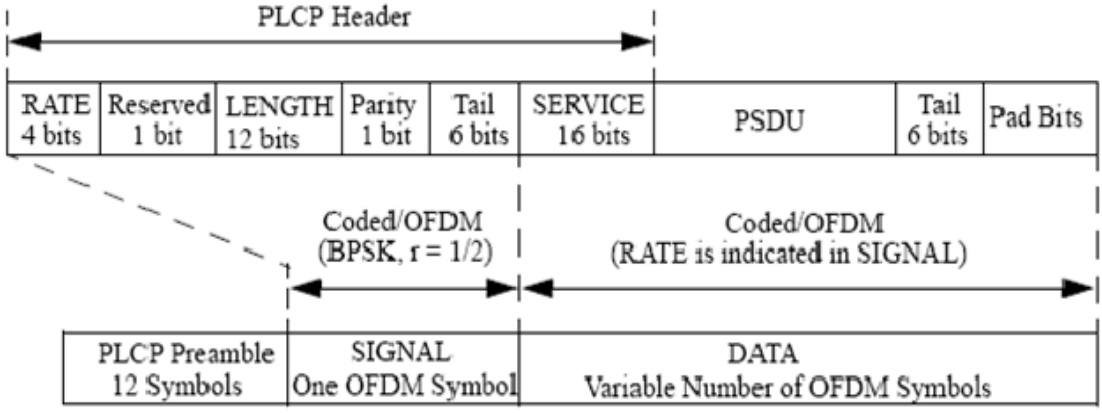

Figure 6. PPDU Frame Format (From Ref [8]).

Each PLCP Protocol Data Unit (PPDU) frame begins with a preamble. Overall, the length of the preamble is equal to the duration of four OFDM symbols. Figure 7 depicts the structure of the short and long preamble training sequences. The first half of the preamble is composed of ten short training sequences. The short preamble's purpose is to provide for signal detection, Automatic Gain Control (AGC) convergence, diversity selection and timing acquisition to name a few of its contributions. The second half of the PLCP preamble consists of two "long training sequences" that aid in enhanced frequency acquisition for the receiver [8]. These two long sequences are preceded by a CP that is formed and appended in the same fashion as with a standard OFDM symbol. The first half of the preamble omits the CP. The short training sequences that compose the preamble are formed using a normalized sequence of QPSK I-Q data, given by

$$\begin{aligned} S(k) = \sqrt{13/6}\{&0,0,1+j,0,0,0,-1-j,0,0,0,1+j,0,0,0,-1-j,0,0,0,-1-j \\ &,0,0,0,1+j,0,0,0,0,0,0,0,-1-j,0,0,0,-1-j,0,0,0,1+j \\ &,0,0,0,1+j,0,0,0,1+j,0,0,0,1+j,0,0\}, \text{ for } -26 \le k \le 26. \end{aligned} \tag{2.4}$$

This sequence is processed by a 64-point IFFT to produce a short sequence of 16 time samples having duration of 0.8 μs:

$$r_S(n) = w_S(n) \sum_{k=-26}^{26} S(k) e^{(j2\pi k \Delta f n)}, \tag{2.5}$$

where $w_S(t)$ is a window function.

Generation of the long sequence is performed in a similar fashion but by using a sequence of BPSK data given by

$$\begin{aligned} L(k) = \{&1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,1,1,-1,-1,1,1,-1,1,-1,1 \\ &,1,1,1,0,1,-1,-1,1,1,-1,1,-1,1,-1,-1,-1,-1, \\ &1,1,-1,-1,1,-1,1,-1,1,1,1,1\}, \text{ for } -26 \le k \le 26. \end{aligned} \tag{2.6}$$

As with the short sequence, the long sequence is processed by a 64-point IFFT operation in the following manner

$$r_L(n) = w_L(n) \sum_{k=-26}^{26} L(k) e^{\left(j2\pi k \Delta f \left(n - T_g\right)\right)} \tag{2.7}$$

with T_g accounting for the 1.6-μs long CP located at the leading edge of the symbol. The duration of each long training sequence after the IFFT is 3.2 μs. Figure 7 illustrates the

format and duration of the preamble structure for the 20-MHz channel spacing option. Additionally, Figure 7 shows the signal field and data OFDM symbols that follow. The signal field contains the data rate and length of the following transmission. Notice that after the nonstandard OFDM symbol format of the preamble, the subsequent data is encapsulated in standard OFDM symbols that are 4.0 μs in duration.

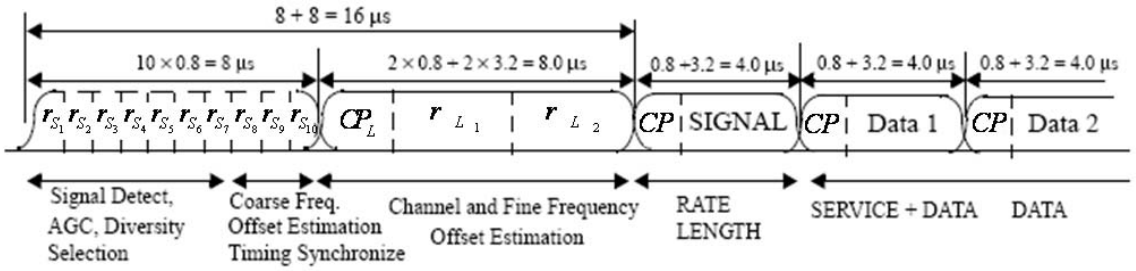

Figure 7. OFDM training Structure for 20 MHz Channel Spacing Option (After [8]).

Now that there is a basic understanding of how the PLCP frames are constructed, a closer look at the construction of the OFDM symbols is warranted. The specific parameters of the three inter channel spacing options available in the 802.11 standard are listed in Table 1. Of note, the number of subcarriers is listed at 52. This number includes only the pilot and data subcarriers. Additionally, there are 11 guard subcarriers and one DC null, which results in a 64-point IFFT forming the basis of each OFDM symbol.

An important feature of the OFDM symbol with respect to this research is the pilot subcarriers. They are positioned at frequency indexed subcarriers -21, -7, 7 and 21 as shown in this sequence:

$$P(k) = \{0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0\}, \text{ for } -26 \le k \le 26, \tag{2.8}$$

which excludes the six negative and five positive guard subcarrier indices but does include the DC null. The non-zero terms in this equation are the permanent values of the pilot subcarriers that are transmitted in each OFDM symbol created by an IEEE 802.11 transmitter.

Information data rate	6, 9, 12, 18, 24, 36, 48, and 54 Mb/s (6, 12, and 24 Mb/s are mandatory) (20 MHz channel spacing)	3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s (3, 6, and 12 Mb/s are mandatory) (10 MHz channel spacing)	1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s (1.5, 3, and 6 Mb/s are mandatory) (5 MHz channel spacing)
Modulation	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM
Error correcting code	K = 7 (64 states) convolutional code	K = 7 (64 states) convolutional code	K = 7 (64 states) convolutional code
Coding rate	1/2, 2/3, 3/4	1/2, 2/3, 3/4	1/2, 2/3, 3/4
Number of subcarriers	52	52	52
OFDM symbol duration	4.0 µs	8.0 µs	16.0 µs
GI	0.8 µs* (T_{GI})	1.6 µs (T_{GI})	3.2 µs (T_{GI})
Occupied bandwidth	16.6 MHz	8.3 MHz	4.15 MHz

Table 1: IEEE 802.11 Major OFDM PHY Parameters (From [8]).

The pseudo-random property of the pilot subcarriers is produced by varying the polarity of the non zero values listed in Equation 2.8. This is accomplished by multiplying each of the permanent $P(k)$ values by a pseudo random sequence. The following equation depicts the 127 element sequence that controls the polarity of the four pilot subcarriers

$$\begin{aligned} n_p(l) = \{&1,1,1,1,\ -1,-1,-1,1,\ -1,-1,-1,-1,\ 1,1,-1,1,\ -1,-1,1,1,\ -1,1,1,-1,\ 1,1,1,1,\ 1,1,-1,1,\\ &1,1,-1,1,\ 1,-1,-1,1,\ 1,1,-1,1,\ -1,-1,-1,1,\ -1,1,-1,-1,\ 1,-1,-1,1,\ 1,1,1,1,\ -1,-1,1,1,\\ &-1,-1,1,-1,\ 1,-1,1,1,\ -1,-1,-1,1,\ 1,-1,-1,-1,\ -1,1,-1,-1,\ 1,-1,1,1,\ 1,1,-1,1,\ -1,1,-1,1,\\ &-1,-1,-1,-1,\ -1,1,-1,1,\ 1,-1,1,-1,\ 1,1,1,-1,\ -1,1,-1,-1,\ -1,1,1,1,\ -1,-1,-1,-1,\ -1,-1,1\}. \end{aligned} \tag{2.9}$$

The signal symbol is the first OFDM symbol in the PLCP frame to include pilot subcarriers. In a Kronecker product-like operation, each pilot subcarrier in $P(k)$ is multiplied by the first element of $n_p(l)$ and processed by the IFFT to produce the signal field symbol. The next symbol will include the second element of $n_p(l)$ being multiplied with the four pilot subcarrier values. This pattern will continue until 127 OFDM symbols have been created. On the 128th OFDM symbol, the sequence $n_p(l)$

will repeat as a cyclic extension for the remaining symbols of the frame. The cyclic reuse of $n_p(l)$ creates a pseudo random sequence for each pilot subcarrier which aids in channel estimation.

2. IEEE 802.11 Medium Access

Medium access in IEEE 802.11 is accomplished using a contention-based approach called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In CSMA, users or stations will listen or "sense" the medium to determine its state (busy or idle) prior to transmitting. The state of the medium and the rules governing the CSMA algorithm will determine whether the station transmits or delays transmission. The foundation of this protocol is formed by partitioning time into slot times. All actions are initiated at the beginning of one of these slots. When a station has data to transmit, it will sense the medium for traffic. If the medium is idle, the station will delay an Inter Frame Slot (IFS) period of time, sense again and transmit if the medium is idle. If the medium is busy, the station desiring to transmit will wait until the medium is idle, delay one IFS, and then enter a random backoff phase before transmitting. The purpose of random backoff phase is to avoid simultaneous transmissions by waiting stations. If a transmission is sent while a station is in backoff mode, the delaying station will pause its backoff timer until the transmission is complete then restart the timer.

In the above discussion, we presented some important features of the IEEE 802.11 WLAN standard. We will now review the IEEE 802.16 standard.

C. IEEE 802.16-2004 SPECIFICATIONS

The description of the IEEE 802.16 specifications [7] will begin with the PHY and briefly cover the MAC functions that are of significance to this thesis. Only downlink specifications of the standard will be addressed as the uplink transmissions are not utilized for the identification and classification purposes in this thesis.

1. IEEE 802.16 Physical Layer

IEEE 802.16 transmissions are structured in a time division multiple access (TDMA) frame based format. Figure 8 illustrates the standard Time Division Duplex (TDD) frame format for an 802.16 transmission. The base element of the 802.16 frame is the physical slot, having the duration

$$t_{ps} = \frac{4}{f_s} \tag{2.10}$$

where f_s is the sampling frequency. The number of physical slots per frame is dependent on the data symbol rate and the frame length. A 10-ms frame, for instance, is comprised of 6667 physical slots, which equates to 138 OFDM symbols for both the DL and the UL at a sample rate of 4 MHz. Figure 8 illustrates the frame structure of a TDD transmission format. TDD is the preferred method of two-way communications within the IEEE 802.16 standard and provides for a down link (DL) and an up link (UL). The DL transmission begins with the preamble followed by the Frame Control Header (FCH). From the FCH on, the rest of the OFDM symbols created are of the standard symbol format which will be explained in the following discussion. The FCH is similar to the signal field of an 802.11 transmission in that it describes the transmission rates and format to the subscriber stations. Next, the DL map and the UL map describe when each subscriber station will listen for their transmissions on the DL and when to transmit their data on the UL.

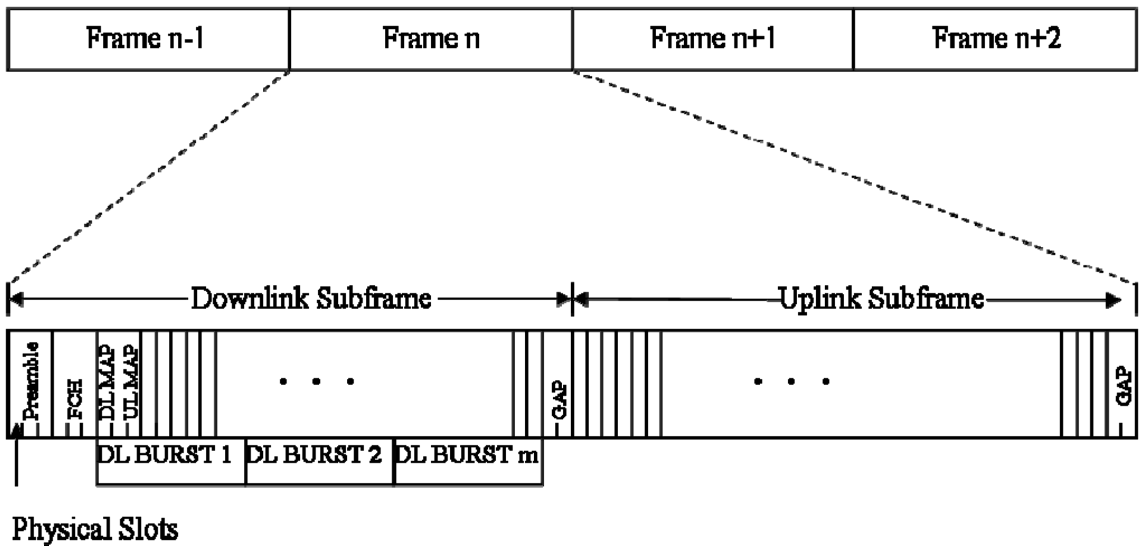

Figure 8. TDD Frame Structure (After [7]).

Each frame begins with a preamble. The preamble is comprised of two OFDM symbols that are constructed from a sequence of QPSK I-Q data as follows [7]

$$
\begin{aligned}
X_{PA}(k) = \{&1-j, 1-j, -1-j, 1+j, 1-j, 1-j, -1+j, 1-j, 1-j, 1-j, 1+j, -1-j, 1+j, 1+j, -1-j, 1+j, -1-j, -1-j, 1-j,\\
&-1+j, 1-j, 1-j, -1-j, 1+j, 1-j, 1-j, -1+j, 1-j, 1-j, 1-j, 1+j, -1-j, 1+j, 1+j, -1-j, 1+j, -1-j, -1-j, 1-j,\\
&-1+j, 1-j, 1-j, -1-j, 1+j, 1-j, 1-j, -1+j, 1-j, 1-j, 1-j, 1+j, -1-j, 1+j, 1+j, -1-j, 1+j, -1-j, -1-j, 1-j,\\
&-1+j, 1+j, 1+j, 1-j, -1+j, 1+j, 1+j, -1-j, 1+j, 1+j, 1+j, -1+j, 1-j, -1+j, -1+j, 1-j, -1+j, 1-j, 1-j,\\
&1+j, -1-j, -1-j, -1-j, -1+j, 1-j, -1-j, -1-j, 1+j, -1-j, -1-j, -1-j, 1-j, -1+j, 1-j, 1-j, -1+j, 1-j, -1+j,\\
&-1+j, -1-j, 1+j, 0, -1-j, 1+j, -1+j, -1+j, -1-j, 1+j, 1+j, 1+j, -1-j, 1+j, 1-j, 1-j, 1-j, -1+j, -1+j,\\
&-1+j, -1+j, 1-j, -1-j, -1-j, -1+j, 1-j, 1+j, 1+j, -1+j, 1-j, 1-j, 1-j, -1+j, 1-j, -1-j, -1-j, -1-j, 1+j,\\
&1+j, 1+j, 1+j, -1-j, -1+j, -1+j, 1+j, -1-j, 1-j, 1-j, 1+j, -1-j, -1-j, -1-j, 1+j, -1-j, -1+j, -1+j,\\
&-1+j, 1-j, 1-j, 1-j, 1-j, -1+j, 1+j, 1+j, -1-j, 1+j, -1+j, -1+j, -1-j, 1+j, 1+j, 1+j, -1-j, 1+j, 1-j,\\
&1-j, 1-j, -1+j, -1+j, -1+j, -1+j, 1-j, -1-j, -1-j, 1-j, -1+j, -1-j, -1-j, 1-j, -1+j, -1+j, -1+j, 1-j, -1+j,\\
&1+j, 1+j, 1+j, -1-j, -1-j, -1-j, -1-j, 1+j, 1-j, 1-j\}
\end{aligned}
\qquad (2.11)
$$

The first preamble symbol consists of four short sequences generated by decimating $X_{PA}(k)$ according to the following equation

$$
S(k) = \begin{cases} 2X_{PA}^{*}(k) & k \bmod 4 = 0 \\ 0 & k \bmod 4 \neq 0. \end{cases} \qquad (2.12)
$$

After the decimation process is performed, the 50 frequency samples generated are concatenated with three other identical groups of 50 samples. This group of 200 samples is now subjected to the IFFT operation with the lower, upper and DC nulls inserted into the appropriate locations. Figure 9 illustrates this process by showing the format of the

samples prior to the IFFT and then the OFDM symbols they form after the IFFT operation. The second preamble symbol is created in a similar fashion. Instead of creating 4 short sequences, two long sequences are produced by the following modulo 2 decimation process:

$$L(k) = \begin{cases} \sqrt{2} X_{PA}(k) & k \bmod 2 = 0 \\ 0 & k \bmod 2 \neq 0. \end{cases} \tag{2.13}$$

These two long sequences are 100 samples in length and are displayed in Figure 9.

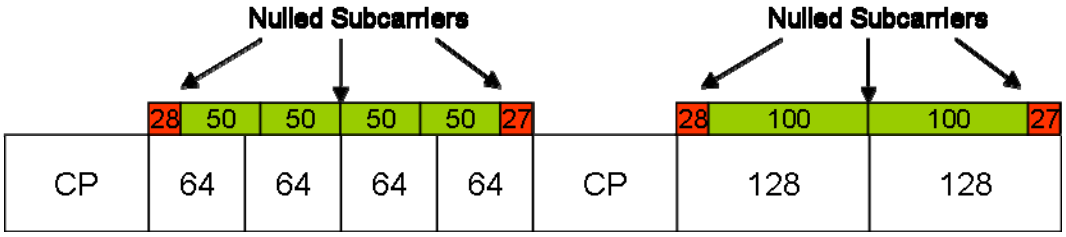

Figure 9. Downlink and Network Entry Preamble Structure (After [6]).

Following the preamble symbols within an IEEE 802.16 frame are the Frame Control Header (FCH), the DL MAP and UL MAP. These control messages are embedded in OFDM symbols that are of the same format as the data carrying symbols. The construction of the data carrying symbols is a key feature of our waveform identification and classification method and will be the next topic of discussion.

An IEEE 802.16 OFDM symbol is constructed in a fashion similar to that of 802.11. The dimensions are the main difference. For instance, the lower and upper guard nulls consist of 28 and 27 subcarriers, respectively. There are eight pilot and 192 data subcarriers. Including the DC null and there are 256 subcarriers in an IEEE 802.16 OFDM symbol. Unlike IEEE 802.11 waveforms, the 802.16 OFDM symbol CP length is not a fixed size. In addition to the 1/4 length CP, which is standard in 802.11, there is the

option of a 1/8, 1/16 and 1/32 length CP with the 802.16 standard. These shortened CP options reduce excessive overhead in the event multi-path fading is not severe enough to justify the a 1/4 CP.

The pilot subcarriers are modulated with a pseudo random binary sequence (PRBS) that is created by the eleven-bit shift register depicted in Figure 10. The output of this PRBS generator is show by the following binary irreducible primitive polynomial [7],

$$w(k) = X^{11} + X^{9} + 1 .$$

The final values that are loaded into the pilot subcarriers prior to the IFFT are determined in the following fashion,

$$\begin{aligned} P_{-88}(k) = P_{-38}(k) = P_{63}(k) = P_{88}(k) = \left(1 - 2w(k)\right) \\ P_{-63}(k) = P_{-13}(k) = P_{13}(k) = P_{38}(k) = \left(1 - 2\overline{w(k)}\right) \end{aligned} \tag{2.14}$$

where the subscript of $P(k)$ represents the corresponding pilot subcarrier frequency offset index and $\overline{w(k)}$ is the compliment of $w(k)$. The pilot subcarriers are loaded with the results of Equation 2.14, OFDM symbol by OFDM symbol.

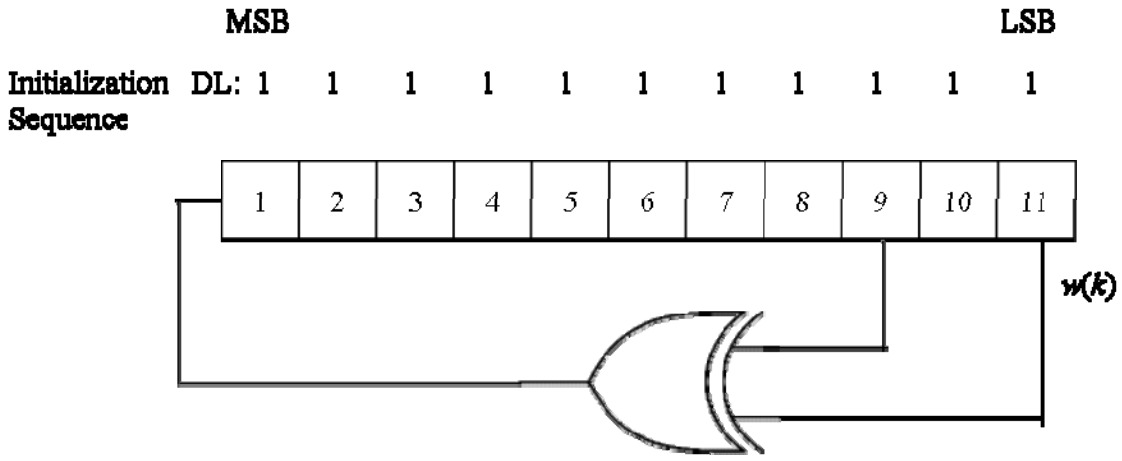

Figure 10. PRBS generator for IEEE 802.16 Pilot Subcarrier Modulation (After [7]).

The baseband modulation schemes used in IEEE 802.16 waveforms are the same four options discussed in Section B.1.

Now, we will take a brief look at the MAC scheme employed by the IEEE 802.16 standard.

2. IEEE 802.16 Medium Access

The MAC scheme utilized by the IEEE 802.16 is Time Division Multiple Access (TDMA), a contention-free approach. This is a very structured scheme where all users communicate over the same channel during an assigned window in time. A centralized controller (the base station) establishes a schedule identifying when each user will listen for their traffic on the DL and when they will transmit on the UL. With low user volume, this method can be quite wasteful. However, as the number of users increases, this method becomes much more efficient than contention based approaches.

Figure 8 depicts the structured nature of a TDMA-based medium access scheme. Each frame begins with a DL transmission that is roughly half of the frame length. Following the preamble, the FCH informs the Subscriber Stations what baseband modulation and coding is being used on their transmission. The DL MAP tells the Subscriber Stations when it is their turn to listen for their data. The UL MAP informs the SS when it is their turn to transmit.

At the beginning of each UL subframe, there is a contention window for initial ranging and new Subscriber Stations network entry. These bursts are short in duration and only consist of a few OFDM symbols. For this reason, we decided to focus on the more structured DL subframe for identifying and classifying 802.16 based waveforms.

The discussion will now cover the fundamentals of cyclostationarity.

D. CYCLOSTATIONARITY

Traditionally, analysis of communication signals embedded in noise has assumed the statistical property of stationarity. Stationarity has several definitions but the most common is Wide Sense Stationarity (WSS). A random process (noisy communication signal) is said to be WSS if its first moment (mean) is constant and its second moment is

a function of the time difference. Even if these two conditions do not hold true for the entire time duration of the signal, small segments of the signal can have a constant mean and time invariant second moment [9]. Cyclostationarity, on the other hand, states that most manmade signals such as communication, radar and sonar signals, to name a few, possess first, second and up to nth-order statistical parameters that vary with time [2]. These variations can be periodic which implies cyclostationarity, or they can possess many different periods, which implies polycyclostationarity. The time varying statistical parameters can manifest themselves as additive sine wave components in the time domain or spectral lines in the frequency domain. Whether these periodicities exist in a cyclostationary form or a polycyclostationary form, they can provide an outstanding method for detecting and identifying different types of communication signals.

Before the discussion goes into how these cyclostationary properties are exploited, a brief review of pertinent terms and properties will be performed. Statistical analysis of random processes is normally performed from two differing approaches: ensemble averages and time averages. The ensemble average approach involves analysis of a sample space of repeated events. Generally, this utilizes density and distribution functions that characterize the ensemble. The time average approach, often referred to as the time-series approach, deals with averages over the time axis of a random signal. The ensemble average is given by

$$E\left\{g\left[X(t)\right]\right\} \triangleq \int_{-\infty}^{\infty} g\left[X(t)\right] f_X(x)\,dx, \tag{2.15}$$

where $E\{\cdot\}$ is the expectation operator and $g(\cdot)$ is a function. The time average operation is given by

$$\hat{E}\left\{g\left[x(t)\right]\right\} \triangleq \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} g\left[x(t+t')\right] dt', \tag{2.16}$$

where $\hat{E}\{\cdot\}$ the time averaging operator. These two statistical analysis approaches can be equated through the principal of ergodicity. If a random process is ergodic, the statistical moments generated via ensemble and time averages are equivalent.

In order to develop the cyclic moments that identify a signal's cyclostationarity, a quadratic transformation is performed. This operation is not unlike the autocorrelation function described as follows

$$R_x(t, t-\tau) = E\left\{x(t)x^*(t-\tau)\right\}. \tag{2.17}$$

To develop the fundamental parameter of second order cyclostationarity, the Cyclic Autocorrelation Function (CAF), the process in Equation 2.17 is performed in conjunction with the Fourier Transformation of $x(t)$ in the following fashion

$$R_x^{\alpha}(\tau) \triangleq \hat{E}\left\{x(t)x^*(t-\tau)e^{-j2\pi\alpha\left(t-\frac{\tau}{2}\right)}\right\}. \tag{2.18}$$

where α is referred to as cyclic frequency and is displayed on a separate axis. This separate axis allows for the visualization of the amount of correlation that exists between frequency shifted versions of the signal. When α is equal to zero, the CAF becomes a traditional autocorrelation function. As α becomes non zero, and cyclic properties possessed by $x(t)$ will be displayed along the cyclic frequency axis.

To better illustrate this cyclic phenomenon, consider an amplitude modulated signal [2]

$$\begin{aligned} x(t) &= \frac{1}{2}a(t)\cos(2\pi f_o t + \theta) \\ &= \frac{1}{2}a(t)\left[e^{j(2\pi f_o t+\theta)} + e^{-j(2\pi f_o t+\theta)}\right]. \end{aligned} \tag{2.19}$$

where $a(t)$ is a random signal whose autocorrelation is nonzero and possesses the following property

$$\hat{E}\left\{a(t)a^*(t-\tau)e^{-j2\pi\alpha t}\right\} \equiv 0 \qquad \text{for all } \alpha \neq 0. \tag{2.20}$$

The next step will be to calculate CAF of $x(t)$ as follows

$$R_x^{\alpha}(\tau) = \frac{1}{4} e^{j(2\pi f_o \tau)} \hat{E}\left\{ a(t) a^*(t-\tau) e^{-j2\pi\alpha t} \right\} + \frac{1}{4} e^{-j(2\pi f_o \tau)} \hat{E}\left\{ a(t) a^*(t-\tau) e^{-j2\pi\alpha t} \right\}$$
$$+ \frac{1}{4} e^{j2\theta} e^{-j2\pi f_o t} \hat{E}\left\{ a(t) a^*(t-\tau) e^{-j2\pi(\alpha - 2f_o)t} \right\} \tag{2.21}$$
$$+ \frac{1}{4} e^{-j2\theta} e^{j2\pi f_o t} \hat{E}\left\{ a(t) a^*(t-\tau) e^{-j2\pi(\alpha + 2f_o)t} \right\}.$$

Applying the property of Equation 2.20 to Equation 2.21 yields

$$R_x^{\alpha}(\tau) = \begin{cases} \frac{1}{4} e^{\pm j2\theta} R_a(\tau) & \text{for } \alpha = \pm 2f_o \\ \frac{1}{2} R_a(\tau) \cos(2\pi f_o \tau) & \text{for } \alpha = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.22}$$

Notice, when $\alpha = 0$, Equation 2.22 is equal the stationary autocorrelation of $x(t)$. Now, we will extend this analysis to the frequency domain to observe the spectral properties of $x(t)$.

According to Gardner [2], by taking the Fourier transform of a continuous time CAF, we produce a function called the Spectral Correlation Density (SCD) as follows

$$S_x^{\alpha}(f) = \sum_{\tau=-\infty}^{\infty} R_x^{\alpha}(\tau) e^{-j2\pi f t}. \tag{2.23}$$

By applying Equation 2.22 to Equation 2.23, we have

$$S_x^{\alpha}(f) = \begin{cases} \frac{1}{4} e^{\pm j2\theta} S_a(f) & \text{for } \alpha = \pm 2f_o \\ \frac{1}{4} S_a(f + f_o) + \frac{1}{4} S_a(f - f_o) & \text{for } \alpha = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.24}$$

Notice that when $\alpha = 0$, the SCD of $x(t)$ reduces to the PSD for this signal.

This chapter covered the fundamentals of OFDM modulation. It described how the IEEE 802.11 and 802.16 standards utilized OFDM modulation to produce communication waveforms and transmit them over a shared wireless channel. Finally,

the concept of cyclostationarity was introduced. The topics discussed in this chapter will be built upon in the following chapters, as we discuss the methods utilized to identify and classify OFDM based wireless network waveforms.

THIS PAGE INTENTIONALLY LEFT BLANK

III. OFDM BASED SIGNAL IDENTIFICATION AND CLASSIFICATION

Currently, most modern wireless data communication networks are utilizing OFDM based modulation techniques of differing configurations. With this in mind, a reliable waveform identification and classification method is becoming necessary. It is especially imperative from an intelligence collection perspective.

Within in the field of cognitive radios, there is much research being conducted on the topic of waveform identification and classification. Cognitive radios maximize usage of idle portions of the electro-magnetic spectrum. When the licensed owner of a reserved frequency band is not actively using this resource, cognitive radios will sense the unutilized portion of spectrum and make use of it [4]. Once the licensed user begins to transmit, the cognitive radio must recognize this traffic and vacate that frequency band. Of the three waveform identification methods mentioned earlier, the trend appears to be toward cyclostationary feature detection. Although matched filter, and energy detection methods hold promise in waveform identification, the matched filter requires a very complex hardware footprint to cover all possible waveforms. Energy detection is susceptible to noise and interfering signals [1]. Cyclostationary feature identification provides for precise and reliable waveform identification and classification.

In this chapter, we will introduce a model to identify and classify IEEE 802.11 and 802.16 waveforms. A method of waveform identification by preamble correlation is discussed. Next, three methods of cyclostationary feature identification are presented and examined in detail. Finally, a computationally efficient algorithm is introduced that produces a SCD estimate of the waveforms analyzed in this work.

A. WAVEFORM IDENTIFICATION AND CLASSIFICATION

Waveform identification and classification can conjure different meanings for different applications. In the context of the thesis, identification shall refer to the determination of which wireless communication standard the waveform is compliant with. Classification shall refer to the determination of which variant of cyclic prefix (CP)

is utilized when constructing the OFDM symbols of the signal being analyzed. In order to achieve the stated objective of this thesis while complying with the above-mentioned definitions, a waveform identification and classification model is proposed as shown in Figure 11. The identification aspect here is limited to the IEEE 802.11 and 802.16-2004 standards.

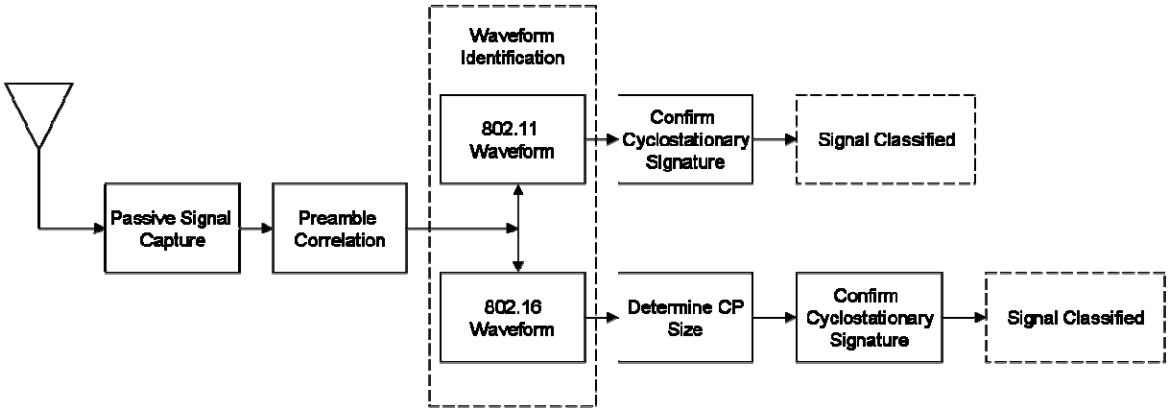

Figure 11. Proposed Waveform Identification and Classification Model.

Once a signal of interest has been captured, it will be down converted to baseband for analysis. Since the signals analyzed in this work are compliant with published IEEE standards, the beginning of a transmission burst or frame will start with a preamble. The purpose of the preamble is to allow the receiver to detect the presence of a compliant signal and for synchronization in both time and frequency. In order to determine if the captured signal is an IEEE 802.11 or an 802.16 waveform, a cross-correlation operation is performed. If the received waveform is identified as 802.11, the identification decision will be confirmed by cyclostationary analysis leading to the classification of the signal. Since the 802.11 standard provides for only one CP length, CP classification is not warranted. When an 802.16 signal is identified, the first step in the signal classification process is to determine the CP of the waveform because of the potential of multiple CP sizes. In order to generate a cyclostationary signature of the waveform, the CP size must be determined and those samples removed prior to FAM estimator. After the CP size has been determined, the cyclostationary signature is extracted and the waveform is classified.

The following discussion will explain the process by which a received waveform is identified through a preamble cross-correlation operation.

B. WAVEFORM IDENTIFICATION BY PREAMBLE CROSS-CORRELATION

This section focuses on the method employed to identify IEEE 802.11 and 802.16 standard compliant waveforms through a cross-correlation process. In this discussion, two signals or processes will be considered. One will be a reference signal $x(t)$ and the other will be a received signal $y(t)$ with additive white Gaussian noise (AWGN). In order to determine if these two signals possess some degree of statistical similarity, the time averaged cross-correlation function is calculated as follows [9],

$$\hat{R}_{XY}(\tau) = \hat{E}\left\{X^*(t)Y(t-\tau)\right\} = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)y(t-\tau)\,dt\,. \tag{3.1}$$

This operation is referred to as the time averaged cross-correlation function. Assuming ergodicity, the corresponding ensemble cross-correlation function can be obtained through the expectation operation,

$$R_{XY}(\tau) = E\left\{X^*(t)Y(t-\tau)\right\}. \tag{3.2}$$

For waveform identification purposes, let $x(t)$ represent an IEEE 802.11 standard compliant preamble sequence consisting of 320 samples, $s_1(t)$, i.e. $x(t) = s_1(t)$. Let $y(t)$ be equal to the same IEEE 802.11 preamble sequence with AWGN, i.e., $y(t) = s_1(t) + n(t)$ where $n(t)$ represents noise and $s_1(t)$ and $n(t)$ are uncorrelated. The cross-correlation of these two processes given by

$$R_{XY}(\tau) = E\left\{\left(s_1^*(t)\right)\left(s_1(t)+n(t)\right)\right\}, \tag{3.3}$$

which reduces to

$$R_{XY}(\tau) = E\left\{s_1^*(t)s_1(t-\tau)\right\}. \tag{3.4}$$

The cross-correlation function for $\tau = 0$ is given by

$$R_{XY}(0) = E\left\{\left|s_1(t)\right|^2\right\}. \tag{3.5}$$

In the case where $x(t)$ is a 320 sample IEEE 802.16 preamble sequence, i.e., $x(t) = s_2(t)$ and $s_2(t)$ and $n(t)$ are uncorrelated, Equation 3.2 reduces to

$$R_{XY}(0) = E\left\{s_2^*(t)\, s_1(t)\right\}. \tag{3.6}$$

In practice, the discrete time cross-correlation function based on Equation 3.1 is computed as follows

$$R_{xy}[l] = \frac{1}{N}\sum_{k=-N}^{N-1} x^*[k]\, y[k+l]. \tag{3.7}$$

Following on the lines of the assumptions made in arriving at Equations 3.5 and 3.6, we have

$$R_{XY}[0] = \frac{1}{N}\sum_{k=-N}^{N} \left|s_1[n]\right|^2 \tag{3.8}$$

when $x(t) = s_1(t)$ and

$$R_{XY}[0] = \frac{1}{N}\sum_{k=-N}^{N-1} s_2^*[n]\, s_1[n] \tag{3.9}$$

when $x(t) = s_2(t)$.

Figure 12 shows a MATLAB generated result of the cross correlation between an IEEE 802.11 standard compliant preamble sequence versus a received signal from an AWGN channel and 802.16 standard compliant preamble sequence versus the same received signal from an AWGN channel. Each sample preamble is comprised of 320 samples, which equates to the entire 802.11 preamble sequence (short and long training sequences). Only the first half of the 802.16 preamble (first four 64 sample sequences with 1/4 length CP) is included in a sample window of this size. In addition, the signal-to-noise ratio (SNR) for the received signal was selected to be 0 dB. Within the context of this work, SNR is defined as

$$SNR = 10\log_{10}\frac{\text{Signal Power}}{\text{Noise Power}}$$

At $l = 0$, Figure 12(a) demonstrates a very high degree of correlation. This indicates a match with the first 320 samples of the received signal. The result displayed in Figure 12(b) indicates little to no correlation.

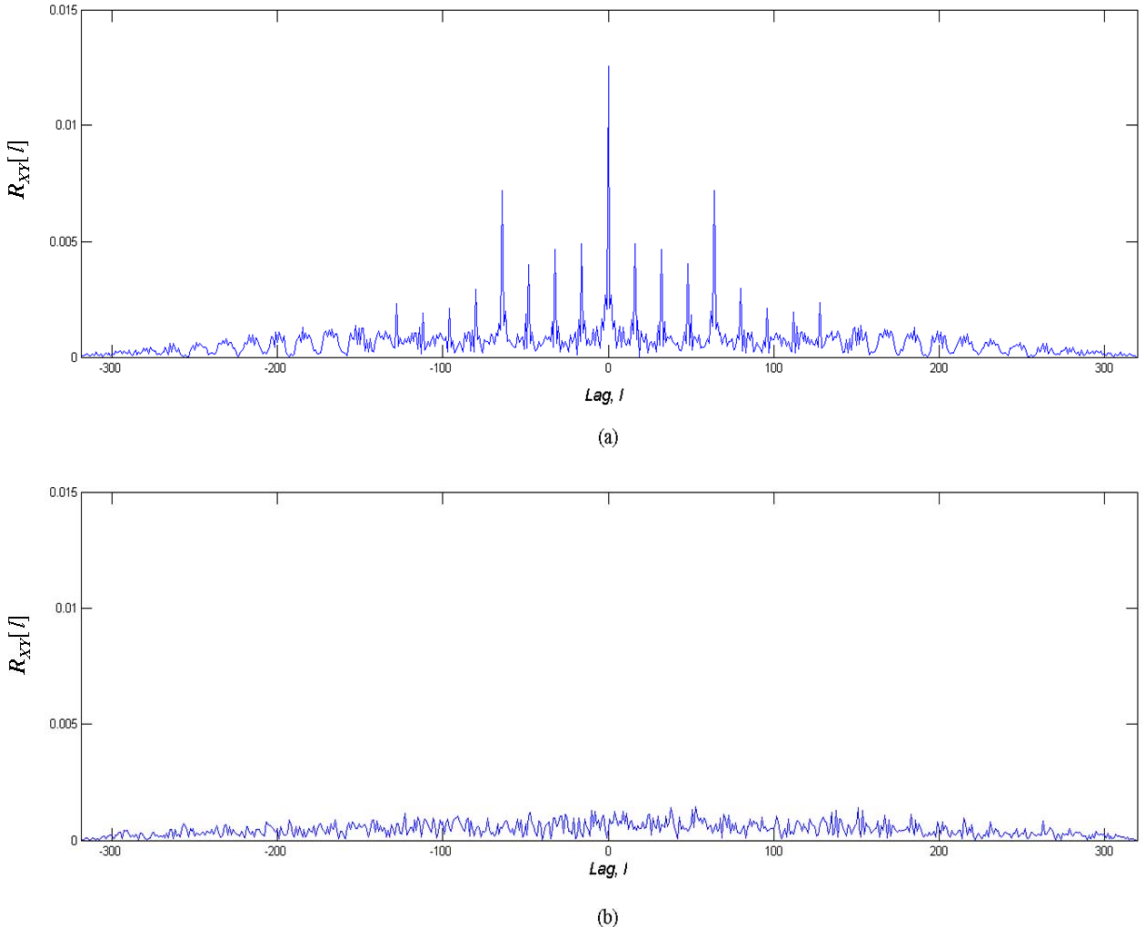

Figure 12. Cross-Correlation of an IEEE 802.11 Preamble Versus (a) 802.11 Sample Preamble, (b) 802.16 Sample Preamble.

Figure 13 depicts the results when an IEEE 802.16 waveform is received from an AWGN channel and subjected to the same cross-correlation process. As is the case with the results displayed in Figure 12, there is an identifiable match and a definitive mismatch. The 802.11 cross-correlation appears to display little to no commonality while the 802.16 preamble cross-correlation indicates a high degree of likeness. For a much more thorough discussion on the process of signal detection and classification via IEEE 802.11 and 802.16 preamble cross-correlation refer to [12].

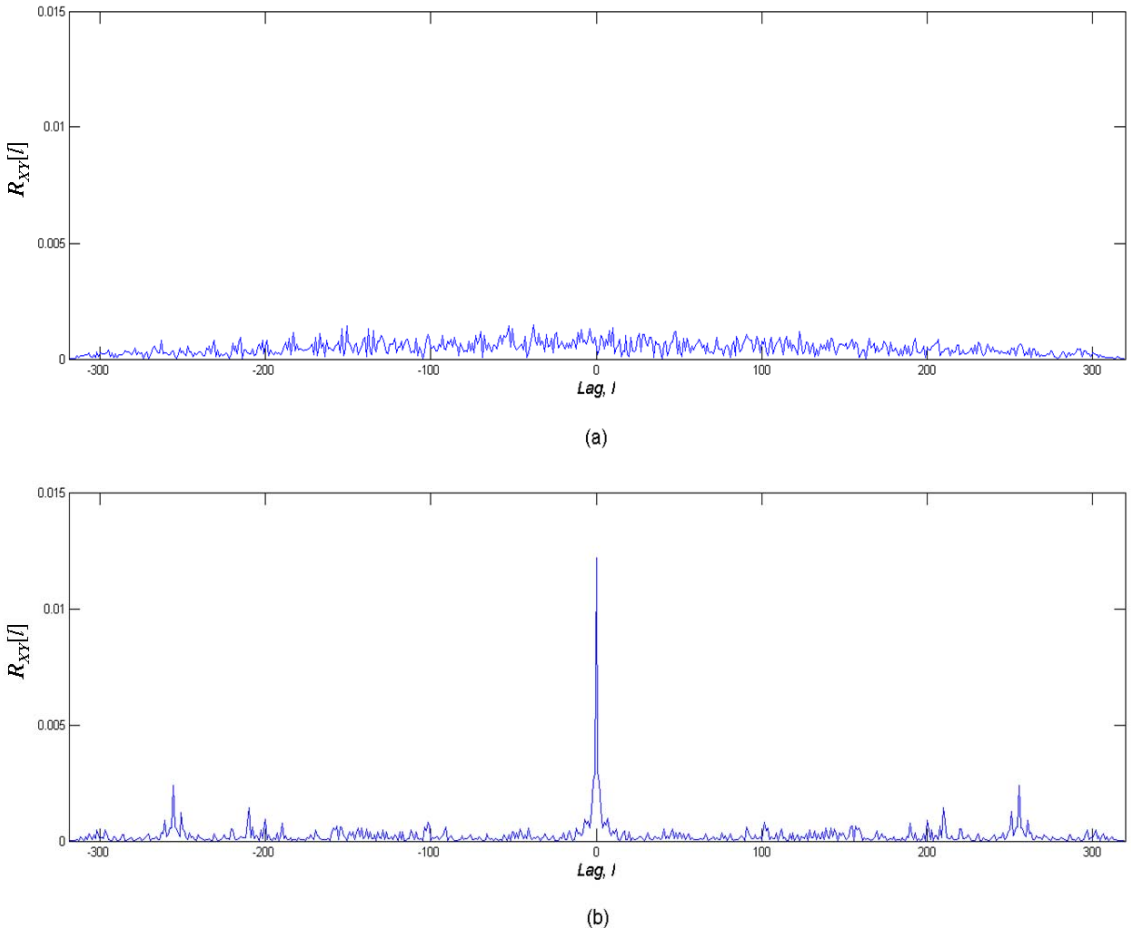

Figure 13. Cross-Correlation of 802.11 Preamble Versus a) 802.11 Sample Preamble, b) 802.16 Sample Preamble.

Now that the preamble cross-correlation operation has identified the type of waveform that was received, the next step will be to confirm the preamble identification. This will be accomplished via cyclostationary feature extraction. In the event an IEEE 802.11 waveform was identified, the cyclostationary feature extraction will confirm the results of the preamble cross-correlation. This process will entail passing the received signal into the FAM SCD estimator and confirming its cyclostationary signature and will be discussed in the next section. If an IEEE 802.16 waveform is identified, cyclostationary feature extraction will be used to determine which CP length was used to construct the OFDM symbols. Once the CP length is determined, this signal will be processed by the FAM algorithm to confirm the waveform identification results of the preamble cross-correlation and the CP classification results.

C. CYCLOSTATIONARY FEATURE EXTRACTION

As mentioned in Chapter I of this thesis, there are three widely known methods of cyclostationary feature extraction in reference to identification and classification of OFDM based signals. The first method exploits the repetitive transmission of the frame preamble [3], the second utilizes or takes advantage of the cyclostationary signature generated by the pilot subcarriers imbedded in each OFDM symbol [1], and the third embeds a predetermined cyclostationary signature into each OFDM symbol, which can be recognized by cyclostationary analysis [4]. In the following discussion, we take a look at each on of these options under consideration.

1. Frame Preamble Cyclostationary Signature

As mentioned in the previous chapter in discussions on the IEEE 802.11 and 802.16 frame construction, all burst transmissions of frame data are preceded by a preamble. The distinct pattern possessed by preambles from each standard allows a receiver to detect the presence of a standard compliant signal, rapidly synchronize to the signal and ultimately recover the data imbedded in the OFDM symbols. At first glance this would appear to be a solid method of waveform identification by cyclostationary analysis [3]. Unfortunately, each standard possesses structural elements that make this approach unsatisfactory.

a. IEEE 802.11 Preamble Considerations

In the IEEE 802.11 standard, frame transmissions do not occur in a set, periodic fashion. In fact, considering that frame lengths vary in size, especially when comparing the downlink and uplink transmissions, and the mechanics of CSMA/CA access scheme, preamble transmissions are essentially a random phenomenon within a WLAN network. This leads to the conclusion that there was no value added by this identification method over a conventional cross-correlation of preamble time samples.

b. IEEE 802.16 Preamble Considerations

The limitations with respect to the IEEE 802.16 standard have to do with the variable length CP appended to the beginning of each OFDM preamble symbol. The

CP tends to disrupt the cyclostationary signature calculation. Since the CP can be one of four possible values depending on channel conditions, as with the 802.11 standard, there is little gain to justify the increased processing required with generating a SCD of the preamble when the cross-correlation of time samples is sufficient.

2. Pilot Subcarrier Cyclostationary Signature

The pilot subcarrier cyclostationary signature method focuses on the spectral lines formed in the cyclic frequencies by the pilot subcarriers. Of the three cyclostationary feature extraction methods investigated in this thesis, pilot subcarrier cyclostationary signature displayed the most promise. Pilot subcarriers are positioned throughout the bandwidth of each standard's OFDM symbols in order to estimate channel conditions [6] [7]. The positioning of these pilot subcarriers is performed in a fashion to ensure the entire spectrum of each OFDM symbol is adequately assessed while minimizing overhead. If too many pilot subcarriers are dedicated to channel estimation, there are fewer data subcarriers dedicated to throughput. With these pilot subcarriers permanently positioned at the same subcarrier frequency offset index and consisting of a pseudo-random BPSK sequence, a cyclostationary pattern is formed.

Ideally, the polarity of the pilot subcarrier BPSK sequence would be fixed. This would generate a robust cyclostationary signature at the pilot subcarrier positions of the principal cyclic frequency axis. Unfortunately, this would create a spectral signature that would manifest itself as spectral lines along the principal frequency axis of the signals' PSD, which is undesirable as it would lead to a high peak-to-average ratio. A high peak-to-average ratio would become an issue when the signal is amplified prior to transmission [5]. For this reason, a pseudo-random BPSK sequence is modulated on to the pilot subcarriers. Unfortunately, the pseudo-random pilot subcarrier sequence requires the SCD of multiple OFDM symbols to be averaged in order to produce a strong cyclic signature for feature extraction.

Figures 14(a) and 14(b) display the SCD and PSD of an IEEE 802.16 waveform, respectively. This result was generated by the FAM SCD estimator, which will be introduced in the next section. The SCD demonstrates a clearly identifiable pilot

subcarrier cyclostationary pattern. This is the signature that is exploited when classifying IEEE 802.11 and 802.16 waveforms, although it is not nearly as robust when the pilot subcarrier sequence is pseudo-random. Unfortunately, Figure 14(b) also exhibits spectral line development at the pilot subcarrier locations of the PSD. This will aggravate the peak-to-average ratio when the signal is passed through a non-linear amplifier prior to transmission.

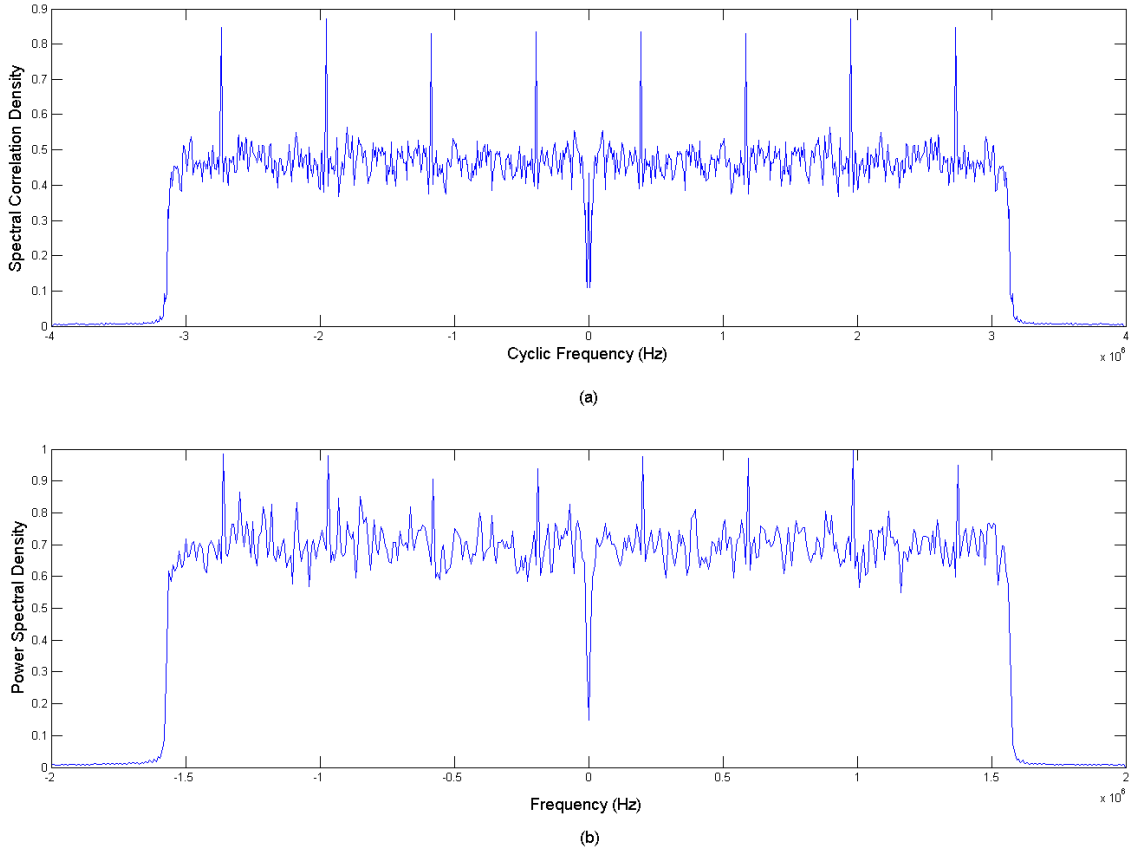

Figure 14. Spectral Line Development: (a) 802.16 Waveform SCD with Constant Pilot Subcarrier Sequence. (b) 802.16 Waveform PSD with Constant Pilot Subcarrier Sequence.

To reduce the peak-to-average ratio, both waveforms utilize a pseudo random BPSK sequence for transmission on the pilot subcarriers. The cyclostationary signature obtained is useful in identifying and classifying OFDM signals.

3. Embedded Cyclostationary Signature

The embedded cyclostationary signature method of OFDM based waveform identification is the topic of much research within the field of cognitive radios. The reason for all the interest in this method is its ability to transmit a unique identification signature. This signature could be employed by licensed users of a range of spectrum to alert non licensed users that the registered user is transmitting. By embedding a unique signature within the OFDM symbols that are carrying the data of the licensed user as underlying periodicities, normal communications can occur with little overhead in terms of bandwidth and power.

This method is similar to the pilot subcarrier sequence method in that the receiver has a priori knowledge of a registered user's signature, which is embedded within its waveform. In order to accomplish this, an operation referred to as subcarrier set mapping is performed [4]. Subcarrier set mapping is accomplished by mapping or routing the same I-Q baseband data inputs to "mirror" locations of the IFFT operation. The term "mirror" refers to the same subcarrier frequency offset indexes with respect to the DC null, i.e., data subcarriers 44 and -44. Subcarrier set mapping produces a cyclic spectral signature that is very distinct for different subcarrier set mapping schemes. Figure 15 is an example of subcarrier set mapping. This is one of many possible configurations of subcarrier set mapping.

A major inefficiency of this scheme is illustrated in Fig. 15. In order to make the signature unique amongst a large population of licensed users, the number of mapped subcarriers must increase. The net effect of increasing the number of mapped subcarriers is the reduction of system throughput. This becomes highly undesirable as the number of mapped subcarriers begins to crowd out the number of data subcarriers.

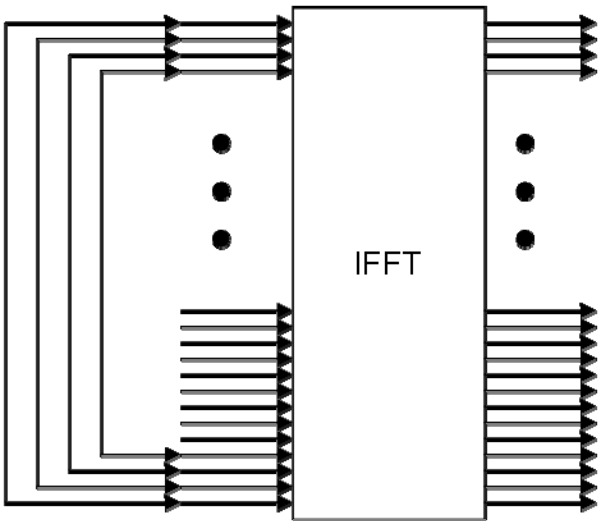

Figure 15. An Example of Subcarrier Set Mapping to Establish an Embedded Cyclostationary Signature.

Within the field of cognitive radios and spectral reuse schemes currently being explored, embedded cyclostationary signature provides a powerful method to positively identify all licensed users. Unfortunately, this method requires a cooperative population of users that will transmit their predetermined signature. This will most certainly not always be the case. In addition, the cyclostationary signature produced will allow for the identification of a specific transmitter (with a priori knowledge of the signature) but not the type of standard compliant waveform produced by the transmitter. With out this knowledge, recovery of the user data would be impossible.

After weighing the advantages and disadvantages of each cyclostationary feature identification method, we determined that the pilot subcarrier cyclostationary signature method would be implemented in this work. The frame preamble cyclostationary signature method is excluded because the cross-correlation process is just as effective with significantly less processing requirements. The embedded cyclostationary signature method, while possessing potential in an identify friend or foe role, is not applicable to identifying transmissions from uncooperative parties; hence, it is not employed in this work.

Next, the discussion will focus on the method utilized to produce the cyclostationary properties of an OFDM based signal.

D. FFT ACCUMULATION METHOD (FAM)

As modern communication system waveforms have increased in complexity, a need arose for computationally efficient methods of producing the SCD of these signals, if their cyclic properties were to be exploited. An algorithm was developed in reference [10] that addressed this need. Two classes of cyclic spectral analysis algorithms were identified in the literature: frequency smoothing algorithms and time smoothing algorithms. While both classes of algorithms are effective at estimating a signal SCD, the time smoothing approach is considered to be more computationally efficient. Within the time smoothing class of algorithms, reference [10] developed two computationally efficient algorithms: the FAM and Strip Spectral Correlation Algorithm (SSCA) while reference [11] developed MATLAB code that implemented the FAM and SSCA algorithms for the detection and identification of cyclostationary signals. After reviewing the results in [11] and testing both algorithms by MATLAB simulation, it was determined that the FAM implementation of the time smoothing approach provided better results for the purpose of this work. This determination resulted from the better resolution provided by the FAM estimator. When considering the number of subcarriers involved in the construction of an IEEE 802.16 OFDM symbol, detailed resolution becomes an imperative feature.

The basics of the time smoothing algorithms for computing the cyclic cross periodogram are described below. The cyclic cross periodogram is given by

$$S_{xy_T}^{\alpha}\left(n,f\right)_{\Delta t}=\frac{1}{T}\hat{E}\left\{X_T\left(n,f+\alpha/2\right)Y_T^{*}\left(n,f-\alpha/2\right)\right\}_{\Delta t} \tag{3.10}$$

where T is the length in seconds of a data tapering window which slides over a signal segment of duration Δt and $X_T\left(n,f+\alpha/2\right)$ and $Y_T\left(n,f-\alpha/2\right)$ are the spectral components of the sampled signals $x(n)$ and $y(n)$, respectively. These spectral components are referred to as complex demodulates and are generated by passing the signals through a narrow-band, band-pass filter. The complex demodulates are produced as follows

$$X_T\left(n,f\right)=\sum_{k=-N'/2}^{N'/2}a(k)x(n-k)e^{-j2\pi f(n-k)T_s} \tag{3.11}$$

and

$$Y_T(n,f) = \sum_{k=-N'/2}^{N'/2} a(k)x(n-k)e^{-j2\pi f(n-k)T_s}. \tag{3.12}$$

where $a(k)$ serves as a data tapering window of length $T = N'T_s$, T_s is the sample duration and N' is size of a sliding point FFT which is described in the next paragraph. Once computed, the complex demodulates are cross-correlated over a time span of Δt seconds.

In an effort to reduce the computational complexity involved with the time averaging process, the FFT can be used to produce the SCD estimate of $x(n)$ and $y(n)$. Development of the complex demodulates remains relatively unchanged with the exception of a sliding N'-point FFT being incorporated to channelize the input samples. Channelization consists of blocks of L samples being selected from the input data and transformed by the FFT in a decimation process intended to increase estimation efficiency. The value of N' is determined via the selected sampling frequency f_s and the desired frequency resolution Δf in the following fashion

$$N' = 2^{\left\lceil \log_2\left(\frac{f_s}{\Delta f}\right)\right\rceil}. \tag{3.13}$$

Next, the complex demodulates are downshifted in frequency to baseband. Now that the complex demodulates have been channelized and reside at baseband, the $Y_T(nL, f_l)$ demodulate is conjugated and multiplied with $X_T(nL, f_k)$ in preparation for the P-point FFT. The value of the P is determined as follows

$$P = 2^{\left\lceil \log_2\left(\frac{f_s}{L\cdot\Delta\alpha}\right)\right\rceil}. \tag{3.14}$$

where $\Delta\alpha$ is the desired cyclic frequency resolution. Now substituting these parameters into Equations 3.11 and 3.12, the FAM estimator using FFT for implementing the time averaging process is given by [10]

$$S_{XY_T}^{\alpha_i+q\Delta x}(nL, f_j)_{\Delta t} = \sum_r X_T(rL, f_k)Y_T^*(rL, f_l)g_c(n-r)e^{\frac{-j2\pi rq}{P}} \tag{3.15}$$

where $g_c(n-r)$ is the Hamming window operation, k and $l = 1,...,N'$ and q is the channel index in the range

$$-\frac{PL}{2N'} \le q \le \frac{PL}{2N'} - 1. \tag{3.13}$$

Figure 16 depicts a block diagram of a generic implementation of the FAM time smoothing algorithm.

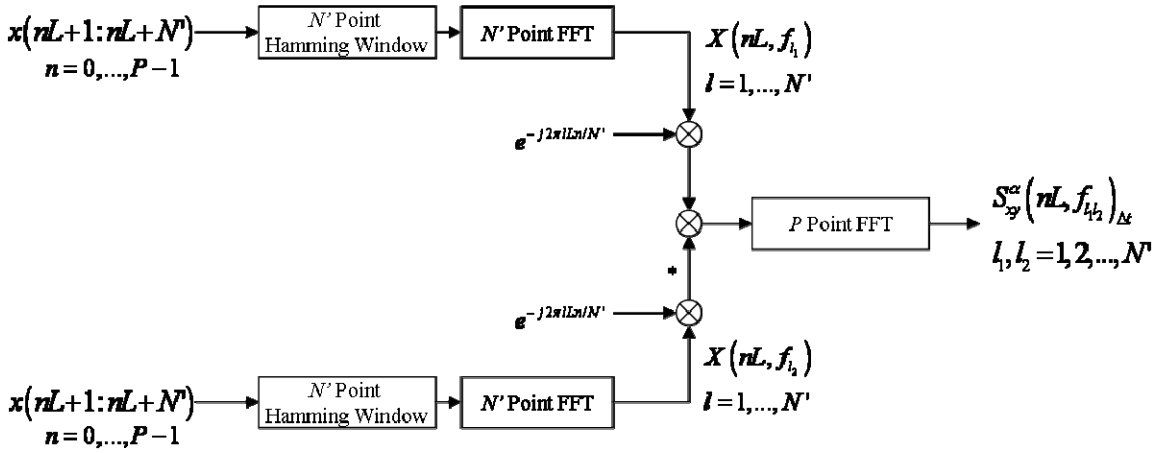

Figure 16. Block Diagram FFT Accumulation Method. (From [11]).

Since the process of generating an SCD estimate has been discussed, the discussion will now focus on how the results of the FAM estimator are presented. Normally, when presenting PSDs of a random signal, the standard method is to display the PSD as a function of frequency on a two dimensional graph. When displaying the SCD of a signal, an additional dimension must be included to represent the cyclic frequency axis. As described in [2], a region of support is formed within a bifrequency plane to represent a signal's SCD function. Figure 17 is an illustration of the bi-frequency plane and the region of support for a lowpass signal. Note that the cyclic frequency bandwidth of the region of support is twice the conventional frequency bandwidth of a low pass signal. The term α represents the cyclic frequency axis and f represents the traditional frequency axis. In order to display the SCD produced by the FAM estimator, a region of support is populated within a $N'+1 \times 2N+1$ array, where $N = PL$ is size of the data vector processed by the FAM. The dashed boundary

represents the dimensions of the array while the shaded area represents the cyclic region of support. Not all the cells of this array contain useful data, only the ones that fall within region of support.

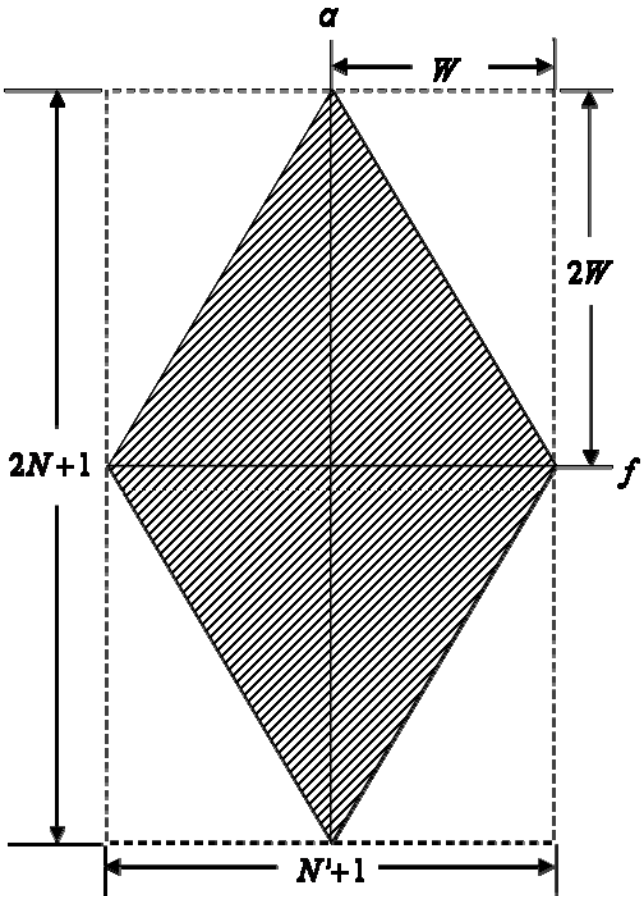

Figure 17. Region of support within the bifrequency plane (From [2]).

The preceding discussion was intended to provide a basic understanding of how the SCD estimate is generated and displayed by MATLAB simulation. A more detailed discussion of the FAM algorithm will be conducted in Chapter IV.

This chapter introduced a method for identifying and classifying OFDM based waveforms. A preamble cross-correlation operation was introduced to identify whether a waveform is compliant with the IEEE 802.11 or 802.16 standard. Next, three methods of cyclostationary feature extraction were introduced. Pilot subcarrier cyclostationary signature identification was determined to be the most applicable method of the three in

achieving the stated goal of this work. Finally, a computationally efficient method of producing the SCD of the OFDM waveforms was introduced and discussed.

Chapter IV will present a MATLAB implementation model to identify and classify OFDM based wireless networking waveforms. This model is the foundation for all simulations conducted throughout the thesis.

IV. BASEBAND SIMULATION RESULTS

In this chapter, a MATLAB simulation is developed to implement the proposed waveform identification and classification scheme presented in Chapter III. The flow of the discussion will begin with a brief description of the simulation model. From this model, a MATLAB implementation is created, which will be the next topic discussed. Each section of the MATLAB implementation is covered in detail. Finally, the discussion will conclude with simulation results, interpretation of the results and conclusions.

A. SIMULATION MODEL

The proposed scheme outlined in Chapter III is implemented by MATLAB code as depicted in Figure 18. The simulation commences with the signal generator producing a single frame of either IEEE 802.11 or 802.16 compliant time samples. This frame will begin with the appropriate preamble followed by a predetermined number of OFDM symbols. The symbols will consist of user data and be constructed as described in Chapter II. Next the vector of time samples passes through an AWGN channel and proceeds to the preamble correlation section.

The noisy time samples are subjected to a cross-correlation operation against standard compliant preamble samples for IEEE 802.11 and 802.16. The results of the cross-correlation operation are utilized to make a decision, based on predetermined criteria, as to whether an IEEE 802.11 or 802.16 waveform is identified. If the predetermined threshold is not met, the simulation ends. Once this decision is made, if an IEEE 802.11 signal is identified it is processed by the FAM to produce the SCD estimate and confirm the preamble correlation results. If an IEEE 802.16 signal is detected, the signal is processed by a test FAM operation to classify the CP length. The test FAM is a specially configured SCD estimator which will identify the CP length of the IEEE 802.16 signal. Since the IEEE 802.11 standard only provisions for a quarter length CP, this process is not necessary for 802.11 signals. With out classification of the CP length in the case of IEEE 802.16, the FAM would not be able to extract the signal's

cyclostationary signature. Once the cyclostationary signature has been produced, the signal is demodulated and compared to the transmitted data string. This, obviously, is to ensure the integrity of the transmission, identification and cyclostationary analysis processes. The next section examines in detail the main MATLAB program.

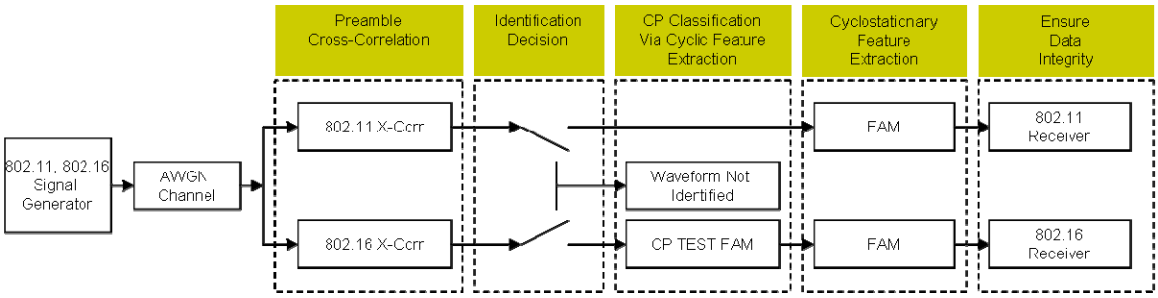

Figure 18. MATLAB Implementation Model.

B. IMPLEMENTATION

The MATLAB implementation of the model depicted in Fig. 18 is performed in a modular fashion. The main program provides a skeletal structure with individual operations being called as functions. All user-defined inputs reside within the main program and are imported to the function routines where applicable. Figure 19 is a flow chart representation of main program. The routine begins with the user selecting the desired waveform to be generated and the desired SNR in the channel. The appropriate waveform generator function is then called to produce one frame of either IEEE 802.11 or 802.16 compliant OFDM symbols with preamble symbols at the beginning. Next the vector of waveform samples has AWGN added to it to simulate the channel. The variance of the AWGN is determined by the SNR selected by the user. At this point in the routine, the preamble of the noisy sample vector is cross-correlated with 320-sample long standard compliant IEEE 802.11 and 802.16 preamble sequences. By meeting or exceeding predetermined identification criterion, the preamble correlation identifies the waveform as being 802.11 or 802.16 compliant. If the identification criterion is not met, the routine terminates as no match is detected.

In the event an IEEE 802.11 waveform is identified, the program will call a FAM SCD estimation function, which will generate the SCD estimate of the waveform. The main program then analyzes the results of the SCD function to confirm the results of the preamble correlation operation.

If an 802.16 waveform is identified, the routine will first classify the CP and then confirm the results of preamble correlation in a similar process as with 802.11 waveforms.

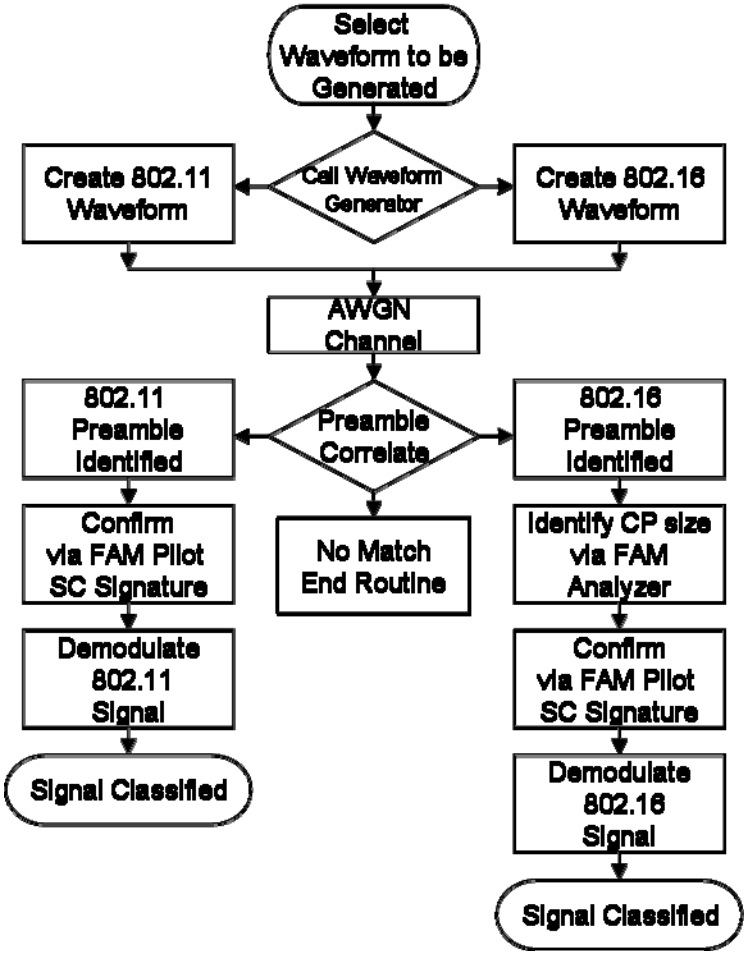

Figure 19. Flow Chart of Main Program.

1. Baseband Signal Generation

In order to produce IEEE 802.11 and 802.16 compliant waveforms for simulation, two MATLAB programs were developed. Generation of each waveform is accomplished via WiFi_BasebandMod.m and WiMax_BasebandMod.m, two separate functions called

by the main program. The first function produces IEEE 802.11 compliant waveforms and the second produces IEEE 802.16 compliant waveforms. Each of these functions follows a similar pattern. Figure 20 illustrates the sequence of events that transpire when the signal generator function is called. First, the signal generator creates a symbol constellation for the appropriate baseband modulation scheme. The baseband modulation scheme used is determined by a logic sequence in the main program and is a function of the user-defined channel SNR. For an IEEE 802.11 waveform, a 0.8-ms down link frame is used. This equates to one preamble sequence, a signal symbol and 195 OFDM symbols being created, each 4 μs in length. An IEEE 802.16 frame length is set at 10 ms but only the downlink portion of the frame is utilized. This equates to two preamble symbols and 67 control and data OFDM symbols.

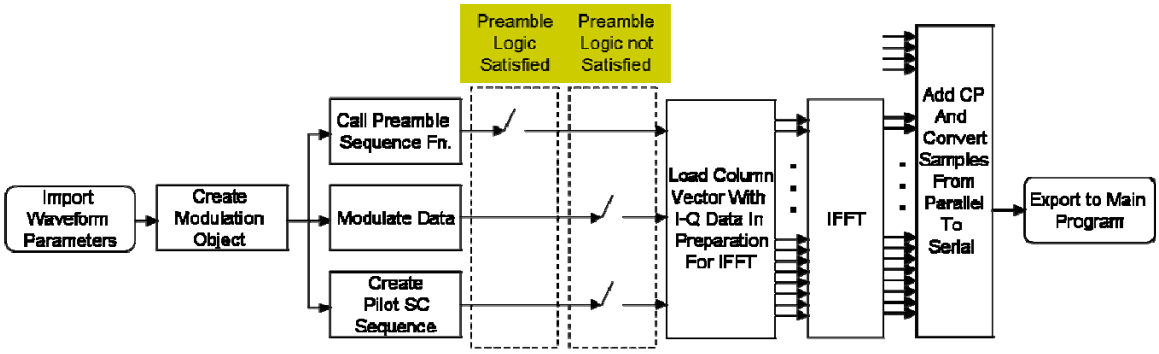

Figure 20. Generic flow of Signal Generator.

To form an OFDM symbol, first a column vector of baseband modulated I-Q data of size equal to the number of data subcarriers is formed. For IEEE 802.11 waveforms the size is 48 and for 802.16 it is 192. The pilot subcarriers are inserted with the corresponding standard defined pseudo-random sequence at the appropriate subcarrier frequency offset index locations. The guard subcarriers (nulls) are then included to complete the vector for the IFFT operation. After the IFFT operation, the CP is appended to the front of the IFFT output values, thus completing the formation of an OFDM symbol.

Data manipulation, to include Forward Error Correction (FEC) coding, data scrambling, and interleaving, are intentionally omitted from both signal generators as they have no impact on the cyclostationarity of the waveform.

Each signal generator has the capability to generate BPSK, QPSK, QAM-16 and QAM-64 baseband modulated waveforms. Figure 21 illustrates the I-Q constellation of a transmitted QAM-64 modulated signal. Of note, the data point located at zero in-phase voltage and zero quadrature voltage represents the DC null. The I-Q values located at (-1,0) and (1,0) coordinates represent the pilot subcarrier BPSK modulated symbols. The remaining 64 I-Q values represent the data symbols carried by the data subcarriers.

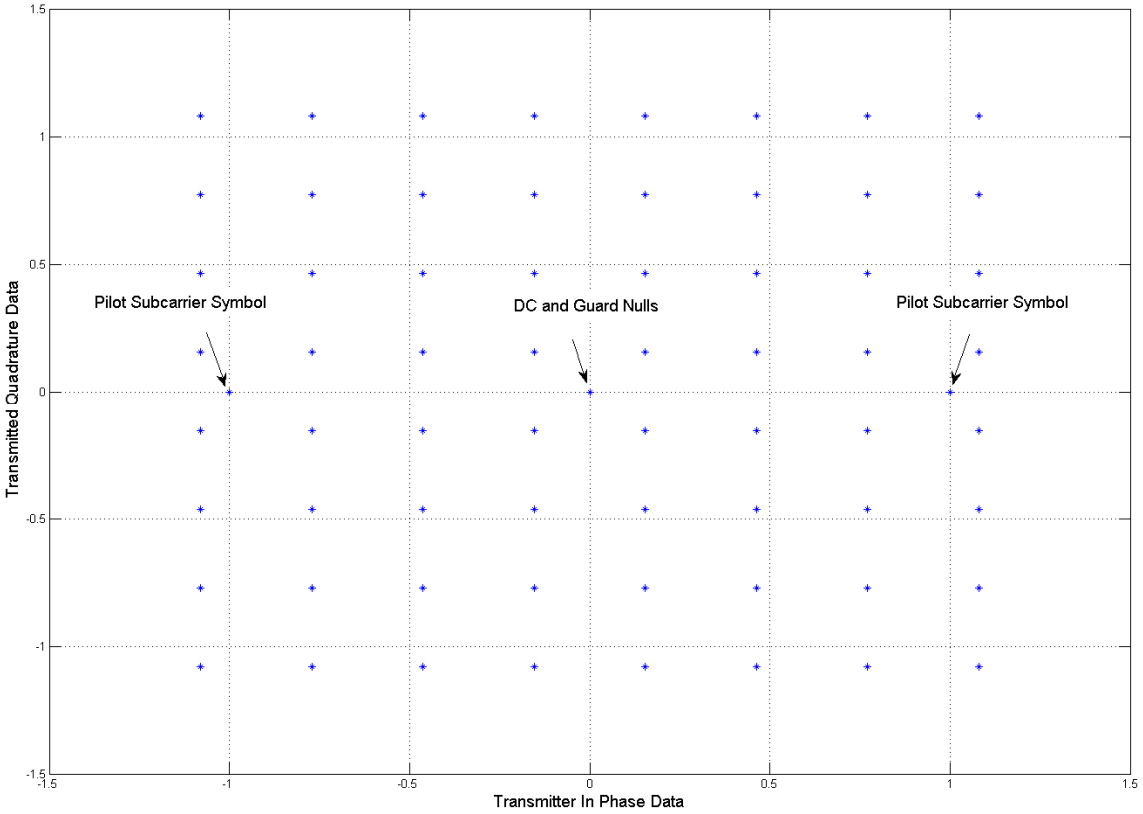

Figure 21. I-Q Voltage Constellation of Quadrature Amplitude Modulation 64 with DC/Guard Nulls and Pilot Symbol Locations.

2. Preamble Cross-correlation Identification

As described in Chapter II, IEEE 802.11 and 802.16 waveforms have their own distinct preamble sequences, which are transmitted at the beginning of each burst transmission. Because each waveform's preamble is unique, it provides a reliable method of identifying the presence of a standard compliant waveform. There are two correlators in this portion of the main program that compare 320 samples of the received waveform with 320 samples of reference waveform. The first correlator's reference

waveform consists of 320 samples of IEEE 802.11 standard compliant data. The second correlator's reference waveform is composed of 320 samples of IEEE 802.16 standard compliant data which consists of the first OFDM preamble symbol with a 1/4 length CP. By using the 1/4 CP length option as the reference waveform, we ensure that the entire preamble sequence will be present in the received waveforms. If the received and reference waveforms match, the cross-correlation process will produce a clearly identifiable peak for $l = 0$ in Equation 3.7. If no peak is present from either correlator, the received waveform is neither IEEE 802.11 nor 802.16 and the program will terminate. Identification criteria will be addressed in the results portion of the chapter.

The next operation performed by the main program will depend on the waveform that is identified. If an 802.11 waveform is identified, the next step will be to confirm the identification by its cyclostationary signature. If the received waveform is identified as 802.16, the CP will next be classified as described in the next section.

3. 802.16 CP Classification

This section explains the process in which the test FAM SCD function determines the CP size of the received signal. Although preamble correlation is an effective method of identifying the waveform received, an IEEE 802.16 signal can not be decoded without knowledge of the CP size. The length of the CP is determined when the network is setup. There are no transmissions that identify the CP length of an IEEE 802.16 waveform, so the passive listener needs a method to classify the CP length. IEEE 802.11 waveforms always use a CP that is ¼ the length of the OFDM symbol. For this reason, IEEE 802.11 identified waveforms are not subjected to the test FAM operation. A method of identifying the CP length is necessary in order to know how many samples to discard at the beginning of each OFDM symbol. In addition, when the CP is not identified and removed prior to cyclostationary analysis by the FAM estimator, the cyclostationary signature of the signal is significantly diminished.

Figure 22 depicts a flow chart representation of the process that classifies the CP of IEEE 802.16 identified signals. After the preamble correlation operation identifies that an IEEE 802.16 waveform is present, the received signal vector is imported into the IEEE

802.16_autofamtest.m function. The IEEE 802.16_autofamtest.m function is a slightly modified version of the FAM SCD function estimator described in Chapter III. Parameters imported to the IEEE 802.16_autofamtest.m are the same as with the FAM function with the exception of the CP size. Since the CP length is unknown to the receiver, the IEEE 802.16_autofamtest.m is run for the four different possible CP sizes.

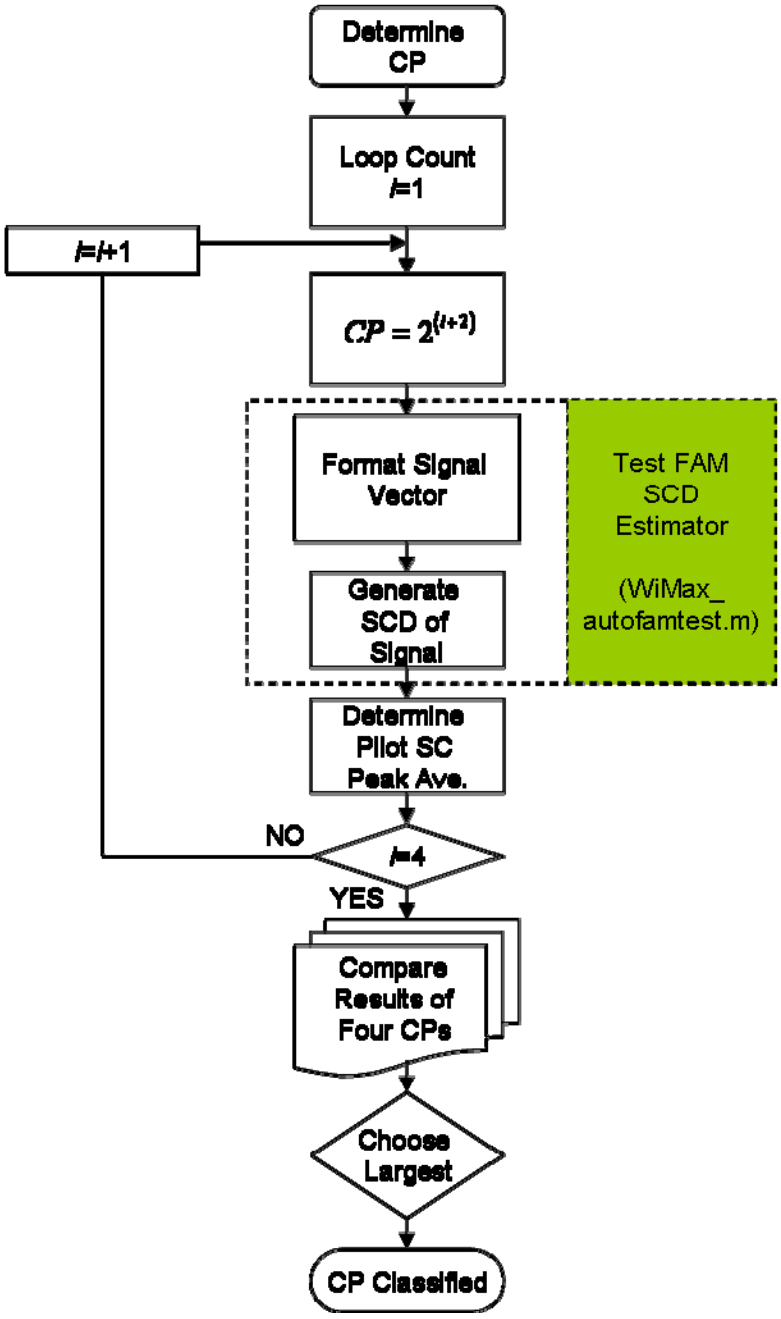

Figure 22. Program Flow to Classify IEEE 802.16 CP length.

The CP classification process begins with the IEEE 802.16_autofamtest.m processing the signal assuming it was generated with a 1/32 length CP. The received signal is imported into the IEEE 802.16_autofamtest.m function along with preselected values for the sampling frequency, frequency resolution, and cyclic frequency resolution, just to name a few. The first step performed by the function is to format the signal vector. This process begins by discarding the preamble samples of the received signal data vector. Within both IEEE 802.11 and 802.16 compliant waveforms, the preamble symbols are nonstandard OFDM symbols, as described in Chapter II. They would tend to diminish any cyclostationary pilot subcarrier signature that is being generated by the estimator. After discarding the preamble samples, the CP of each OFDM symbol is discarded by removing the first eight samples of every successive 264 sample blocks throughout the input data vector. This removes a 1/32 CP amount of samples from each symbol in preparation for FAM processing. Once this process has been performed on the entire vector, the data is processed by the test FAM estimator as mentioned in Chapter III with one notable exception. In order to develop the cyclic pattern of the pilot subcarriers, more than one OFDM symbol must be processed. By processing multiple symbols, the spectral lines produced by the pilot subcarriers' cyclostationary signature will become more pronounced.

The data subcarriers are modulated with random I-Q data, and thus, possess no deterministic reoccurring pattern. The pilot subcarriers are modulated with pseudo-random BPSK data and possess a small degree of repetition. In order to exploit and strengthen this marginal cyclic pattern, SCD results from multiple blocks of samples are averaged.

The averaged SCD values from the FAM operation are displayed as shown in Figure 23, which depicts the principal cyclic frequencies along the x axis. When a robust pilot subcarrier cyclostationary signature is generated, identifiable spectral lines will be displayed along this axis. These spectral lines will be located at the respective pilot subcarrier position throughout the cyclic spectrum of the signal The shaded portion of the $S_{xx}^{\alpha}(f)$ matrix in Figure 23 is the region of support for the SCD function of the

processed signal within the 257×1025 array (IEEE 802.16 SCD matrix dimensions). Values within this region represent SCD magnitude estimates.

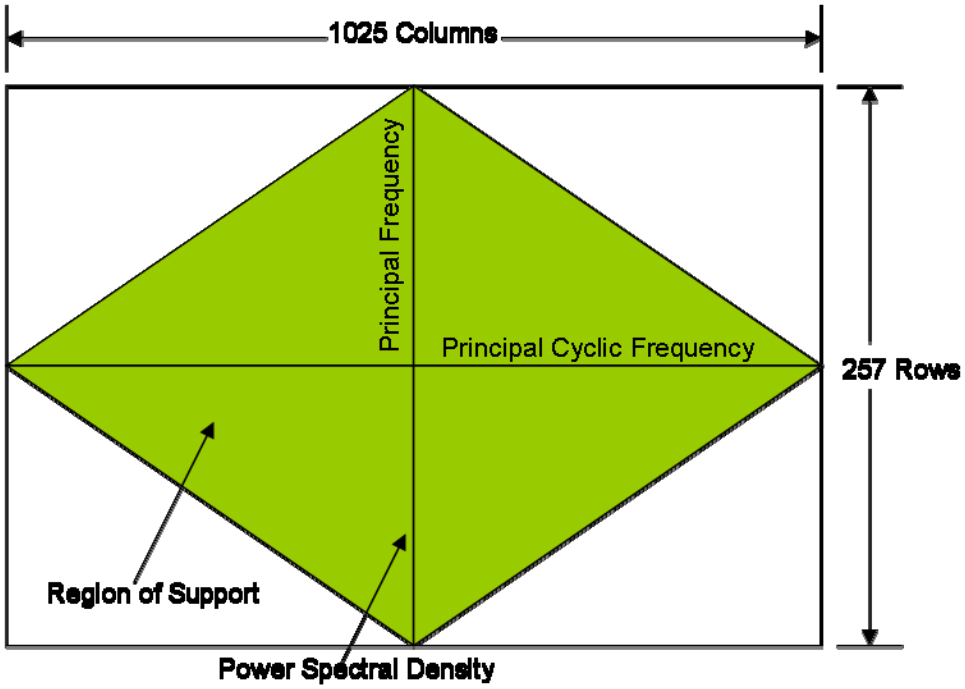

Figure 23. Matrix Description of FAM SCD function estimator Output, $S_{xx}^{\alpha}(f)$.

Since the row containing the principal cyclic frequency values has been identified, the individual cells containing the pilot subcarrier spectral lines now need to be located. As mentioned in Chapter II, IEEE 802.16 compliant waveforms employ a 256 point IFFT to create each OFDM symbol. With the principal cyclic frequency row having 1025 cells, it was determined that *floor*(1025/4) or approximately 4 cells represented a subcarrier. This methodology is utilized to determine an approximate location of the eight pilot subcarrier spectral line locations. Next, the average of the magnitude of these eight spectral lines is determined. There is one cell in this row vector for each possible CP length. The CP length that generates the most robust cyclostationary signature will produce the largest average pilot subcarrier magnitude.

Cyclostationary feature identification is now used to confirm the results generated by the preamble correlation process.

4. Cyclostationary Waveform Identification

Although the received waveform was identified by preamble correlation, the next portion of the model will confirm that result through cyclostationary feature identification. In the case of a received IEEE 802.16 waveform, it has already been subjected to cyclostationary identification when the CP was classified. Now a technique similar in nature to the CP classification process will be utilized. The following discussion will be broken down into two sections. One section will cover the specifics of IEEE 802.11 cyclostationary identification and the second will describe 802.16 specifics.

a. IEEE 802.11 Cyclostationary Feature Extraction

This step of the implementation model is to confirm the preamble cross-correlation waveform identification results produced by the correlators using cyclostationary feature extraction. The process is quite similar to the CP identification discussed in the previous section. The function WiFi_autofam.m is called to produce the SCD function of the 802.11 waveform. The settings used to control the base line parameters of the FAM SCD estimator are imported when the function is called. Some of the parameters imported to the function include: signal vector *famin*, sampling frequency f_s, frequency resolution Δf and cyclic frequency resolution $\Delta\alpha$. As with the CP classification process, the signal vector data is first formatted by discarding the preamble and signal symbol samples at the beginning of the frame. Next the CP is removed from every OFDM symbol throughout the remainder of the frame. Then the formatted signal vector is processed by the FAM SCD estimator. The SCD function, $S_{xx}^{\alpha}(f)$, is averaged over multiple OFDM symbols.

When simulating the IEEE 802.11 waveform, an inter channel spacing factor of 20 MHz was used. Optimal values for Δf and $\Delta\alpha$ where determined to be 156,250 Hz and 78,125 Hz, which were ½ and ¼ the inter subcarrier spacing distances, respectively. These parameters generated output array dimensions for $S_{xx}^{\alpha}(f)$ of 129×513. Row 65 of this array contained the principal cyclic frequency values, to

include four pilot subcarrier spectral lines. The magnitudes of the pilot subcarrier spectral lines are then averaged in the following fashion,

$$P_{pilot} = \frac{1}{4} \sum_{i \in \{\text{pilot indices}\}} S_{xx}^{i}(0) \tag{4.1}$$

where $\{pilot\ indices\} = \{-21, -7, 7, 21\}$. In addition to averaging the pilot subcarriers, the data subcarrier's magnitudes are also averaged as shown

$$P_{data} = \frac{1}{48} \sum_{i \in \{\text{data indices}\}} S_{xx}^{i}(0) \tag{4.2}$$

where $\{data\ indices\} = \{-26:22, -20:-8, -6:-1, 1:6, 8:20, 22:26\}$. The guard and DC subcarriers are omitted as their inclusion would have drastically altered the data subcarrier average.

Now that the average peak magnitudes for the pilot and data subcarrier cyclic spectral lines have been determined, they are compared. The metric used for classification is threshold γ given by

$$\gamma = \frac{P_{pilot}}{P_{data}}. \tag{4.3}$$

For classification implementation a 10% margin was chosen leading to the following classification criteria

$$\begin{aligned} H_1 &: \gamma > 1.1 \\ H_0 &: \gamma < 1.1 \end{aligned} \tag{4.4}$$

where H_1 is a positive classification of an IEEE 802.11 waveform and H_0 is a null hypothesis.

The next section will discuss the IEEE 802.16 specifics of cyclostationary feature extraction.

b. IEEE 802.16 Cyclostationary Feature Extraction

The process of cyclostationary feature extraction employed on 802.16 compliant waveforms is the same as the 802.11 process with one exception; different settings are used to control the base line parameters of the FAM SCD estimator, IEEE

802.16_autofam.m. IEEE 802.16 simulations utilized a transmission bandwidth of 3.5 MHz. This bandwidth required a sampling frequency of 4.0 MHz, which is determined in the following fashion,

$$f_s = \left\lfloor \frac{8}{7} \cdot \frac{W}{8000} \right\rfloor 8000 \tag{4.5}$$

where W is equal to the bandwidth of the signal. The optimal frequency and cyclic frequency resolution were determined to be the inter subcarrier spacing of 15,625 Hz for Δf and half this value, 7813 Hz, for $\Delta\alpha$. As with the 802.11 process, the row containing the principal cyclic data is identified and the location of the eight pilot subcarrier spectral lines identified. These magnitudes were averaged as follows

$$P_{pilot} = \frac{1}{8} \sum_{i \in \{\text{pilot indices}\}} S_{xx}^{i}(0), \tag{4.6}$$

where $\{data\ indices\} = \{-88, -63, -38, -13, 13, 38, 63, 88\}$. The data subcarriers where averaged in a similar fashion as shown

$$P_{data} = \frac{1}{192} \sum_{i \in \{\text{data indices}\}} S_{xx}^{i}(0), \tag{4.7}$$

where $\{data\ indicies\} = \{-100:-89, -87:-64, -62:-39, -37:-14, -12:-1, 1:12, 14:37, 39:62$

$64:87, 89:100\}$. These values are subjected to the same classification criteria discussed in the previous section.

A brief discussion of the receiver section of the implementation model will now be conducted.

5. Baseband receiver

This step of the implementation model is to ensure the data transmitted is recoverable. Although this step is not vital to the identification and classification process, it does lend credibility to the signal generator and allows for the observation of the channel effects on the signal. Both IEEE 802.11 and 802.16 data recovery operations are similar and will be briefly reviewed.

The receiver operation is relatively simple compared to the transmitter operation. The first step the receiver performs is to discard the preamble. Next, in a symbol by symbol fashion, the CP is removed. At this point, it becomes apparent why proper CP classification is critical for an 802.16 waveform. Without classification of the CP, the receiver would not know the proper number of samples to discard for each symbol CP and the data represented by the OFDM symbol samples would become unrecoverable. The I-Q data is now processed by an FFT operation in preparation for baseband demodulation. Next the guard and pilot subcarrier data is discarded. Once the baseband demodulation is performed, the recovered binary sequence is compared to the original to determine bit errors. Figure 24 is an illustration of a received QAM 64 constellation with the channel introducing a SNR of 20 dB. A bit error rate of 0.004618 was determined after data recovery and comparison in this case. For comparison, the theoretical probability of bit error is calculated from [14],

$$P_B \approx 2\frac{\left(1-\frac{1}{\sqrt{M}}\right)}{\log_2\left(\sqrt{M}\right)}Q\left[\sqrt{\left(\frac{3\log_2\sqrt{M}}{(M-1)}\right)\frac{2E_b}{N_o}}\right] \tag{4.8}$$

where M is equal to 64 and E_b/N_o is equal to 20 dB. The result of the calculation yields 3.7×10^{-3}, which is reasonably close considering that Equation 4.8 is an approximation and the sample size of the simulation was 156,672 bits long, not an adequately large sample size.

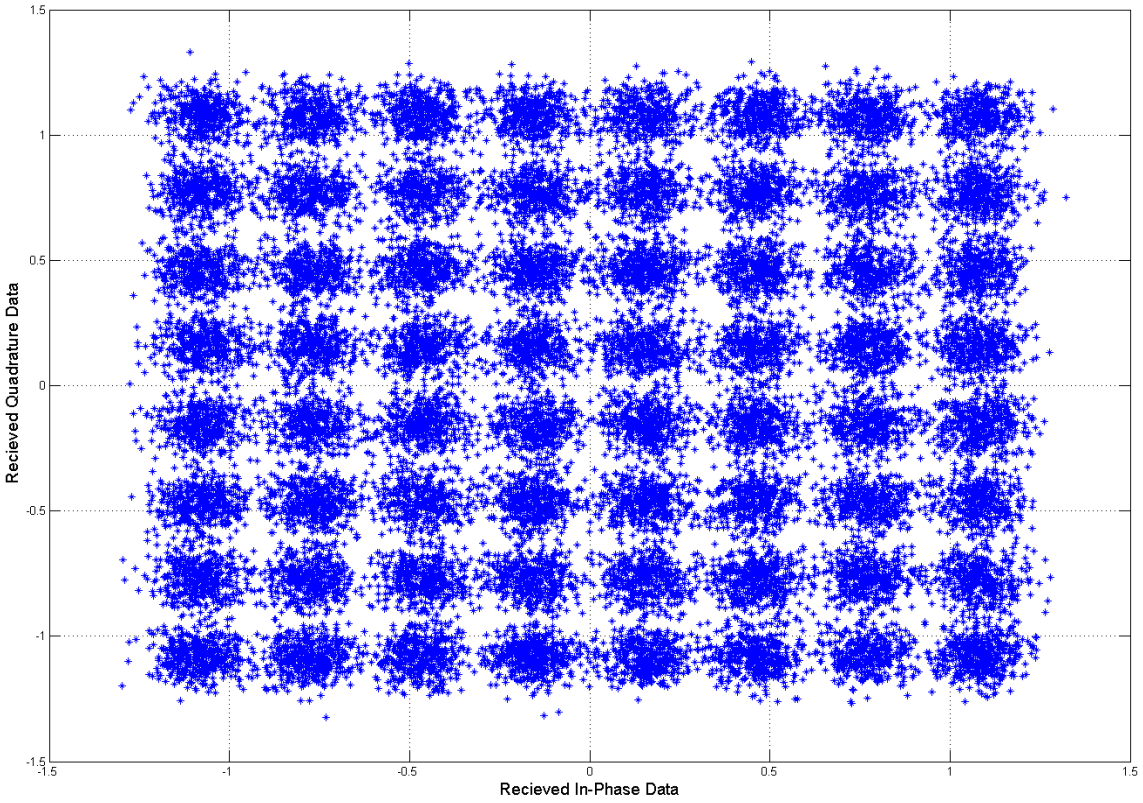

Figure 24. Received I-Q Voltage Constellation of Quadrature Amplitude Modulation 64 with Channel Conditions Simulating a SNR of 20 dB.

The next section with cover the results of the implementation model, as well as a few simulations to help identify some limitations of the model that we need to be aware of.

C. RESULTS

The discussion now shifts to the results of the implementation model. The first results addressed will be the preamble correlation findings, followed by CP classification and finally the cyclostationary feature extraction.

1. Preamble Cross-Correlation

Figures 12 and 13 from Section A of Chapter III display the output of each correlator with both IEEE 802.11 and 802.16 waveforms passed through them at an SNR of zero dB. This process was extended to multiple SNRs in order to identify when the

results would become overwhelmed by the power noise and to identify a point at which a decision threshold could be set. Figure 25 illustrates the results of a simulation run to determine the minimum SNR that still produced reliable results from each correlator when an IEEE 802.11 preamble sequence is received. In addition to running this simulation at 21 different SNRs, all SNR iterations were repeated 150 times in order to average out the effects of noise. Two input waveforms were averaged by each correlator for this simulation, the received 802.11 waveform and a white Gaussian noise only waveform. Although the 802.11 correlator produces the largest average peak at all SNRs tested, a significant separation between this curve and the others occurs at -14 dB, where the matched correlator output approaches a steady value of 0.01256 while the uncorrelated output continues to decrease towards 0.00145. The correlation with white Gaussian noise input waveform approaches zero in both correlator outputs. This agrees with the results of Equation 3.7 where $s_1(t)$ is noise only.

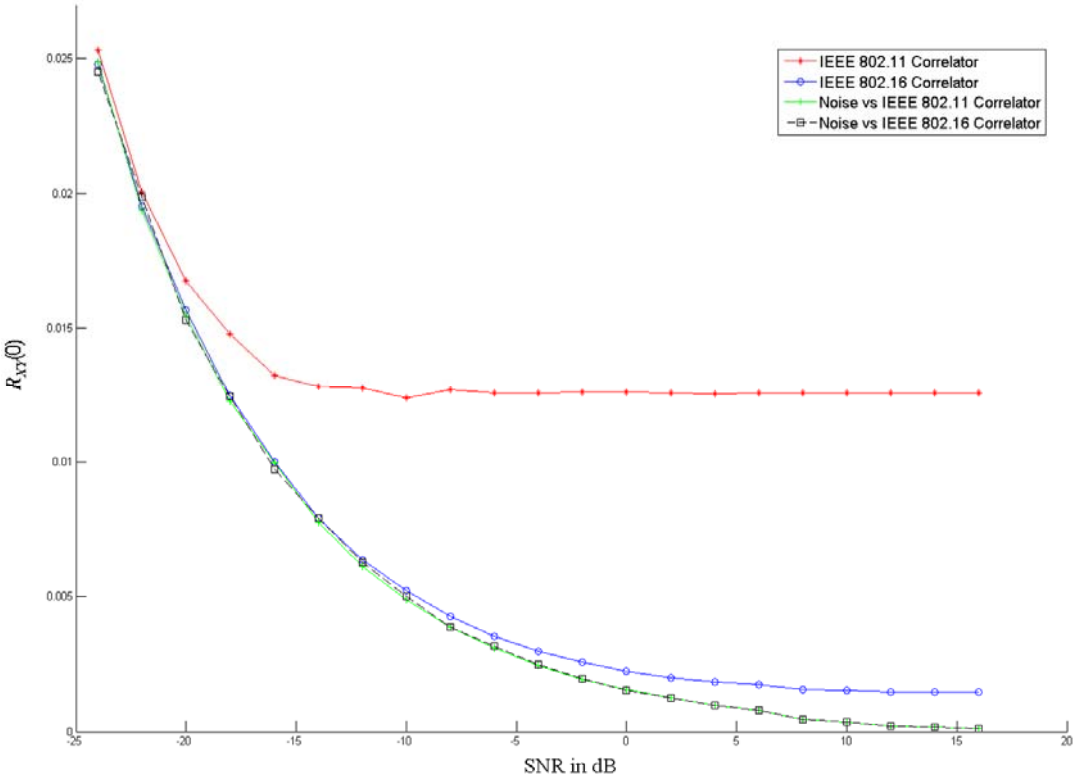

Figure 25. Preamble Correlation Results of an IEEE 802.11 Received Waveform versus SNR Averaged Over 150 Runs.

Figure 26 illustrates the results of the same simulation when an 802.16 waveform or noise only waveform is received. As with the 802.11 results, a significant separation between the 802.16 correlation results and the remaining three results occurs at −14 dB. The 802.16 results approach a value of 0.01011, with the matched correlator output, while the unmatched results decrease to 0.001931. Again, the noise only waveform results decrease toward zero correlation as SNR increases.

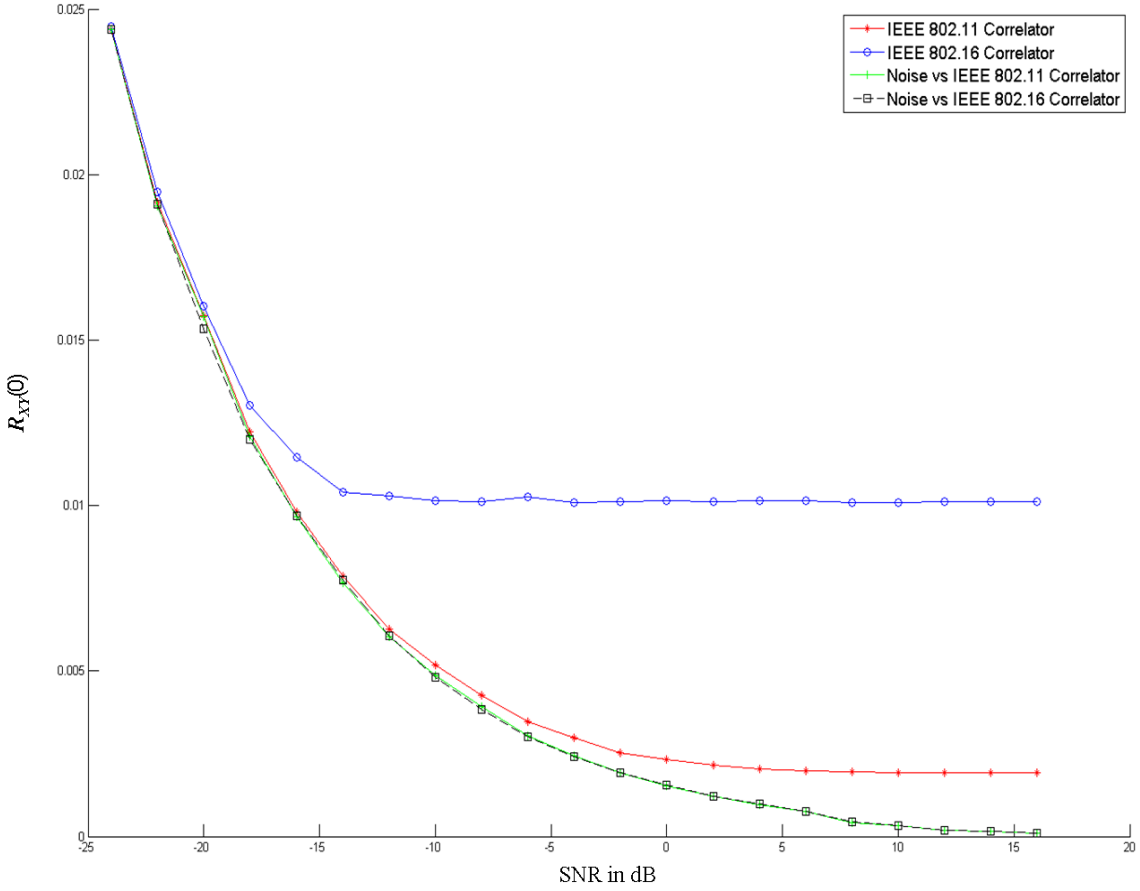

Figure 26. Preamble Correlation Results of an IEEE 802.16 Received Waveform versus SNR Averaged Over 150 Runs.

The purpose of this simulation was to demonstrate that preamble cross-correlation identification is effective at determining which waveform is received at an SNR range well below the receiver's capability to recover the data. Based on visual observation of the simulation results displayed in Figures 25 and 26, without a detailed application of detection theory, a threshold of 0.009 was chosen as the decision metric for waveform

identification. By choosing 0.009 as the threshold, reliable identification decisions would be a reasonable expectation down to an SNR of -14 dB in both cases.

Next, the discussion moves on to the results of the CP classification portion of the simulation.

2. IEEE 802.16 CP Classification

In order to validate the CP classification operation discussed in Section B-3, a simulation was performed to determine the minimum number of OFDM symbols that were required to be processed for a correct classification. There are four possible outcomes of this routine, one for each CP length that is possible: 1/4, 1/8, 1/16, 1/32. The test FAM SCD estimator tested a 1/4 length CP waveform as described in Section B-3 against all four possible outcomes 200 times. The results were obtained for an integer number of OFDM symbols from 3 to 24 symbols. To measure the effectiveness of the classification, the metric used is CP classification percentage: number of successful tests out of the total number of trials (200 in this case). Figure 27 illustrates the results of this simulation. Notice, by 20 OFDM symbols, the test FAM will classify the correct CP 100 percent of the time.

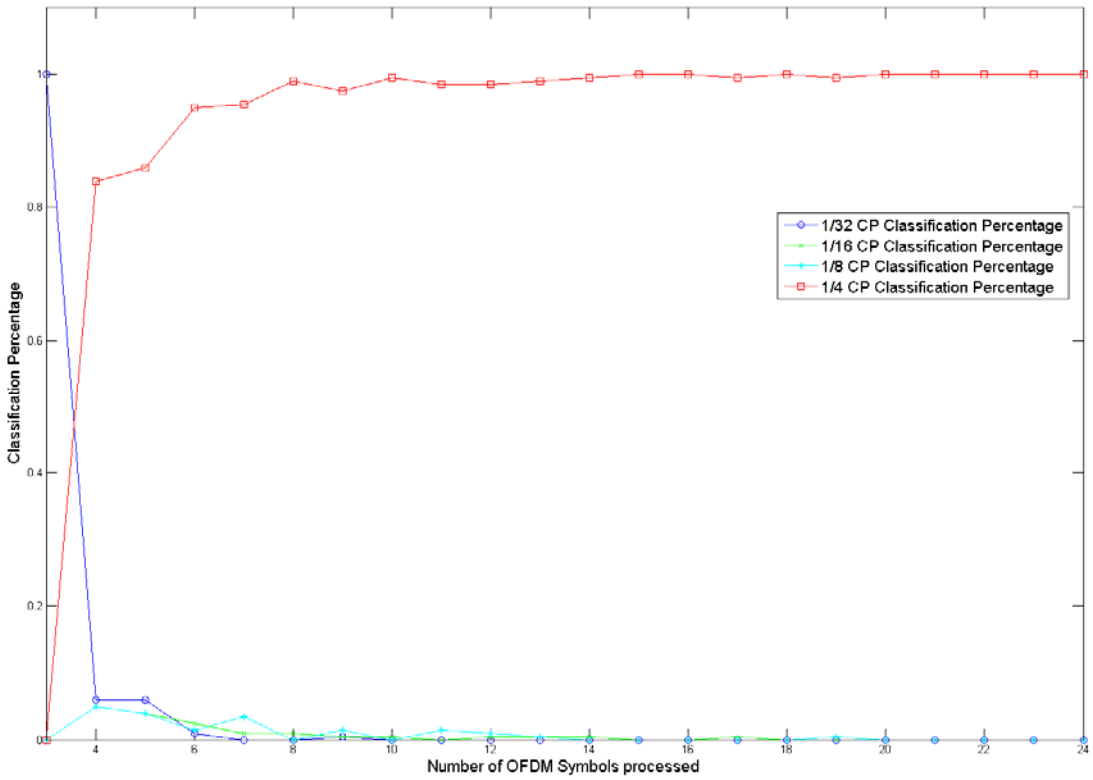

Figure 27. CP Classification Percentage versus the Number of OFDM Symbols Processed by the Test FAM SCD Function Estimator. Averaged Over 200 Runs.

3. Cyclostationary Feature Extraction

The final step of the implementation model for both waveforms is to confirm the identification and CP classification operations by cyclostationary feature extraction. The cyclostationary signatures of each waveform are presented as well as the results of two simulations conducted to determine minimum threshold requirements. The first simulation is designed to identify the minimum SNR where the cyclostationary feature extractor still produces reliable results. The second simulation is designed to determine the minimum number of OFDM symbols that needs to be processed by the FAM SCD estimator to produce reliable results.

a. IEEE 802.11

Figures 28-30 illustrate the cyclic signature exploited by cyclostationary feature extraction. These figures are generated using the MATLAB function

WiFi_autofam.m. Figure 28 is a surface plot of the positive cyclic frequencies of $S_{xx}^{\alpha}(f)$. The left portion of the graph represents the PSD of the waveform with the positive cyclic frequency realizations extending to the right. The pilot subcarrier peaks can be seen protruding well above the data subcarrier floor in arrays of three then two and finally one. To ensure the pilot subcarrier peaks are clearly identifiable, the results averaged over 120 OFDM symbols.

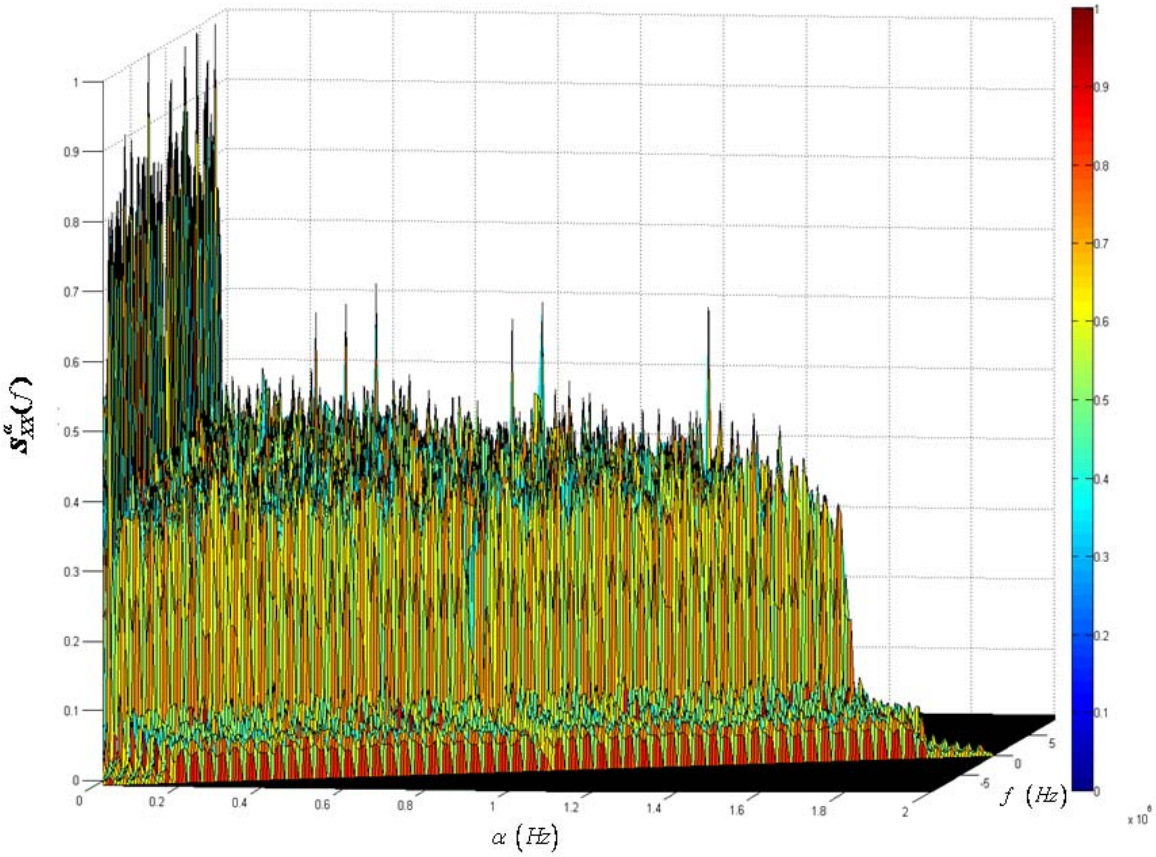

Figure 28. Surface Plot Representation of an IEEE 802.11 Waveform's $S_{xx}^{\alpha}(f)$, Averaged Over 120 OFDM Symbols.

Figure 29 consists of the profile plots that display SCD versus the principal cyclic frequency axis and PSD versus the principal frequency axis. The pilot subcarriers spectral lines are not identifiable in Figure 29(a) and clearly discernable in Figure 29(b).

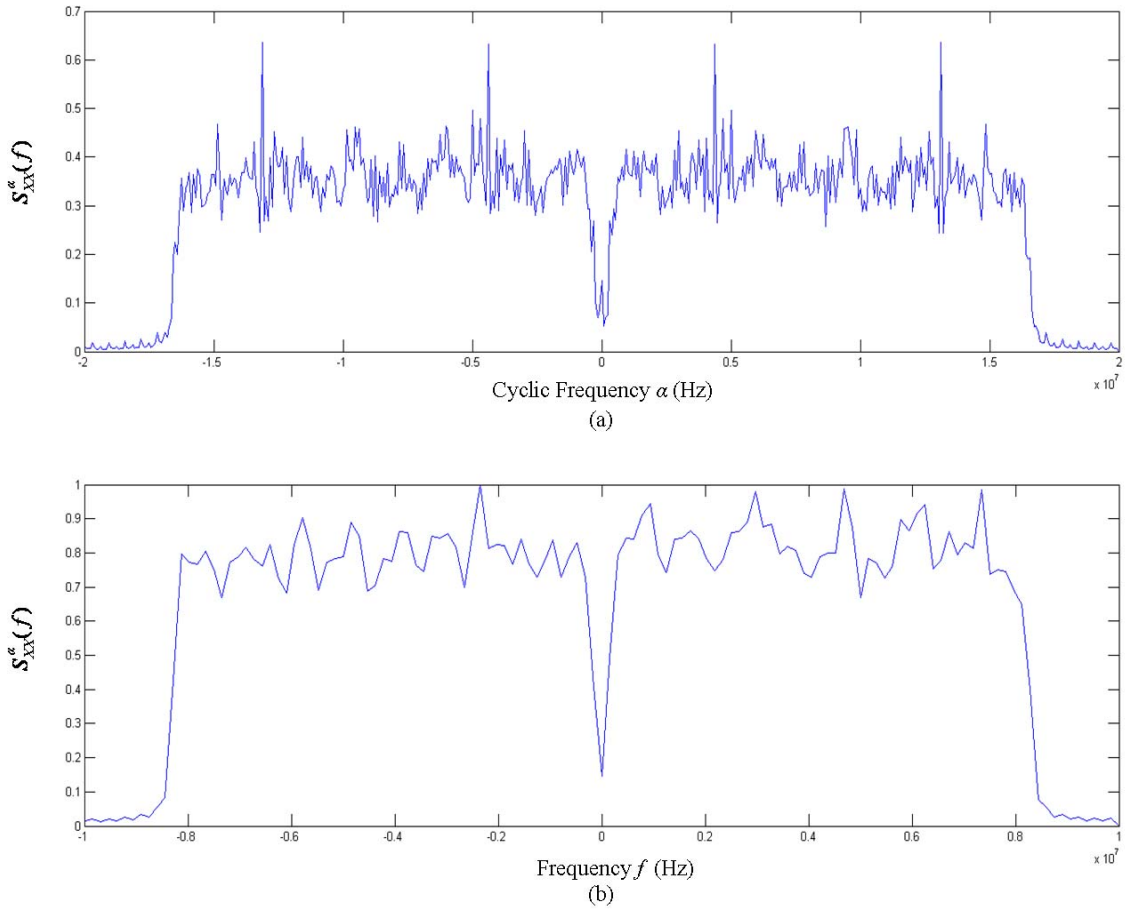

Figure 29. Profile Plots: Magnitude of $S_{xx}^{\alpha}(f)$ versus a) Cyclic Principal Frequency and b) Frequency for IEEE 802.11 Waveform. Results Averaged Over 120 OFDM Symbols.

Figure 30 provides contour plot of $S_{xx}^{\alpha}(f)$ at a level of 0.5. Unlike Figure 28, the contour plot includes the negative cyclic frequencies. The PSD is clearly identifiable, located at the zero cyclic frequency range with the DC null located at the center of the graph. As the cyclic frequency components increase in magnitude, the pilot subcarrier locations appear in a symmetric fashion.

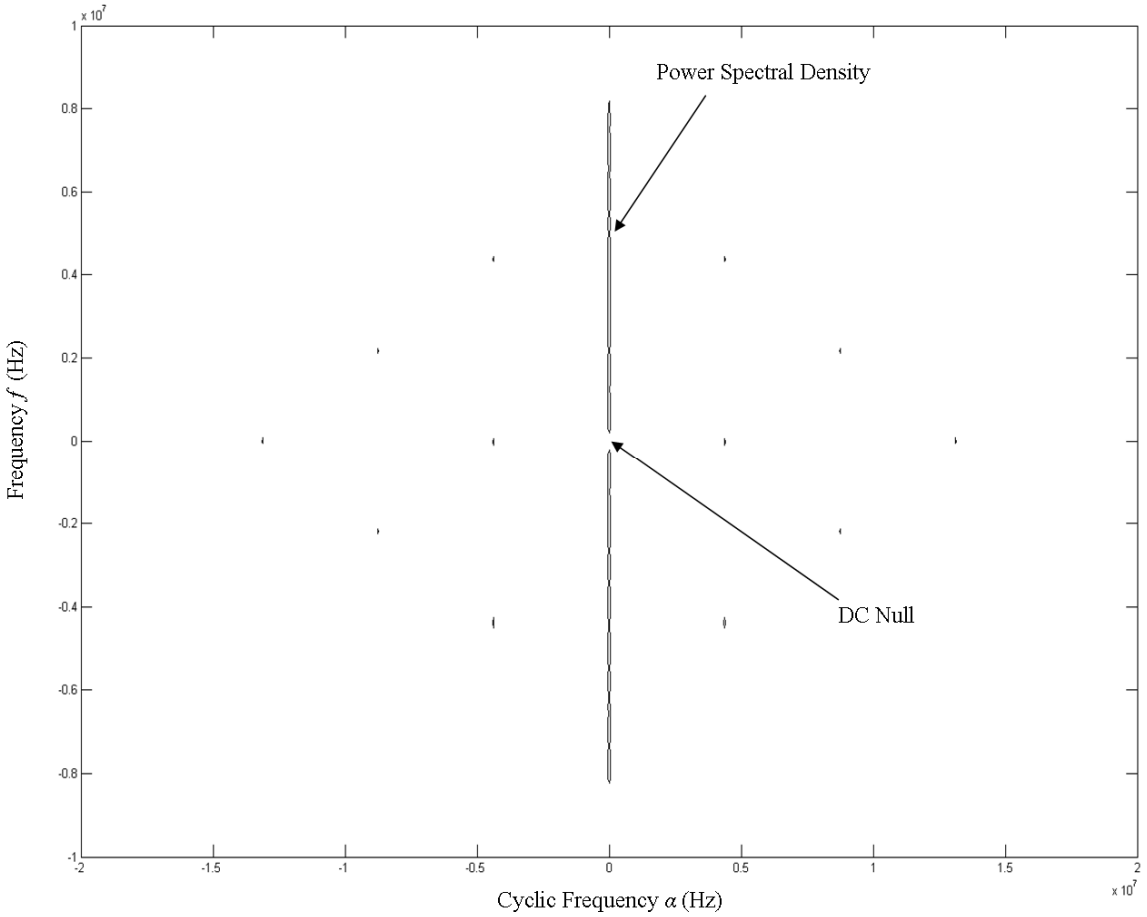

Figure 30. Contour Plot of an IEEE 802.11 Waveform's $S_{xx}^{\alpha}(f)$ Magnitude. . Results Averaged Over 120 OFDM Symbols.

Figures 31 and 32 were generated from simulations that were intended to determine the effects of the SNR and number of OFDM symbols on the subcarrier SCD estimates. Figure 31 shows the effect of SNR on the peak pilot and data subcarrier magnitudes. The results are obtained by averaging results from 150 simulation runs. At roughly -15 dB, the pilot subcarrier SCD peak value clearly began to emerge from the data subcarrier values.

Figure 32 shows the effect of the number of OFDM symbols used on the SCD estimates of pilot subcarriers and data subcarriers. The results are intended to determine how many OFDM symbols are required to produce a clearly identifiable separation between the pilot and data subcarrier SCD peaks.

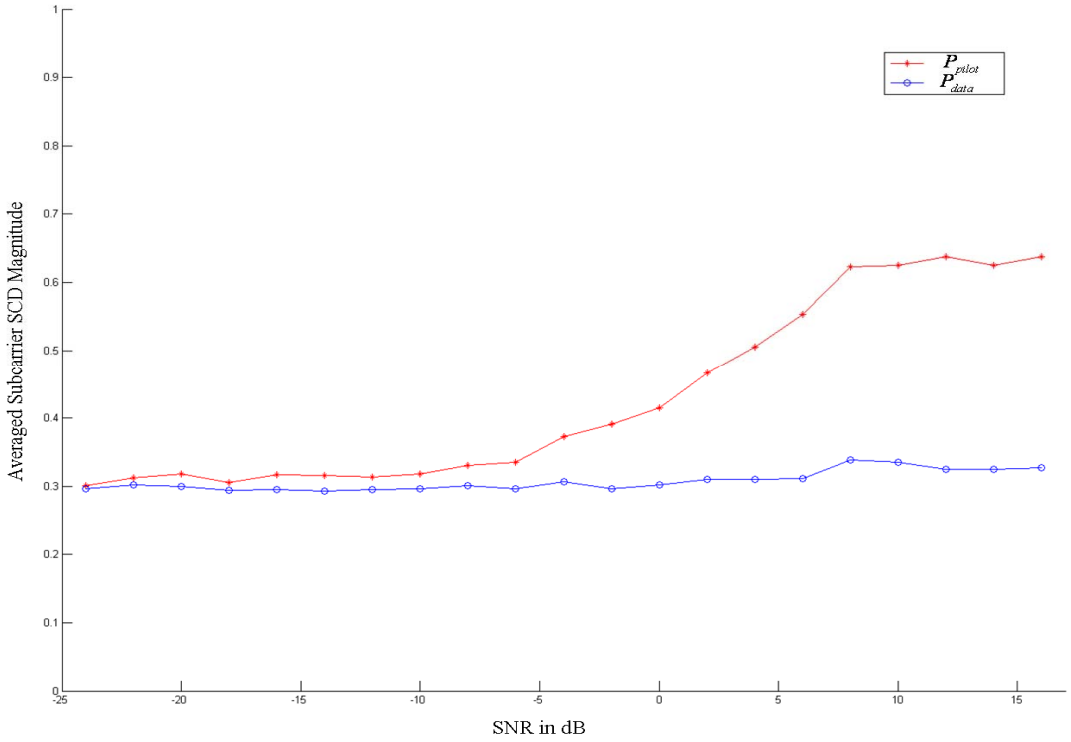

Figure 31. IEEE 802.11 Subcarrier SCD Values versus SNR.

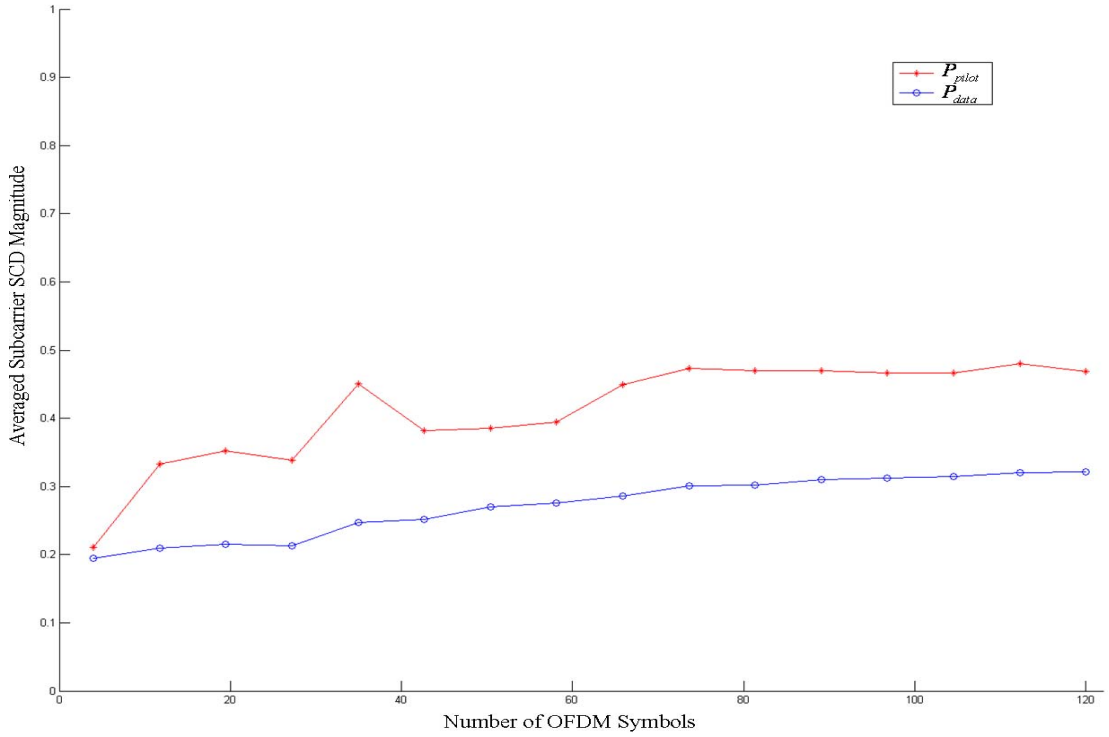

Figure 32. IEEE 802.11 Averaged Subcarrier SCD Values versus Number of OFDM Symbols Processed by the FAM SCD Function Estimator.

b. IEEE 802.16

Plots similar to those obtained for 802.11 waveforms were generated for an 802.16 waveform: a surface plot, a profile plot and a contour plot. These plots were generated from a waveform possessing a 1/32 length CP with results averaged over 120 OFDM symbols using the MATLAB function IEEE 802.16_autofam.m. Figure 33 depicts seven rows of pilot subcarrier cyclic spectral lines. Of these, only four possess pilot subcarrier spectral lines that reside along the principal cyclic axis ($f = 0$). The peak magnitude of the averaged SCD values are normalized to one. Further detail of the principal axis is provided in Figure 34.

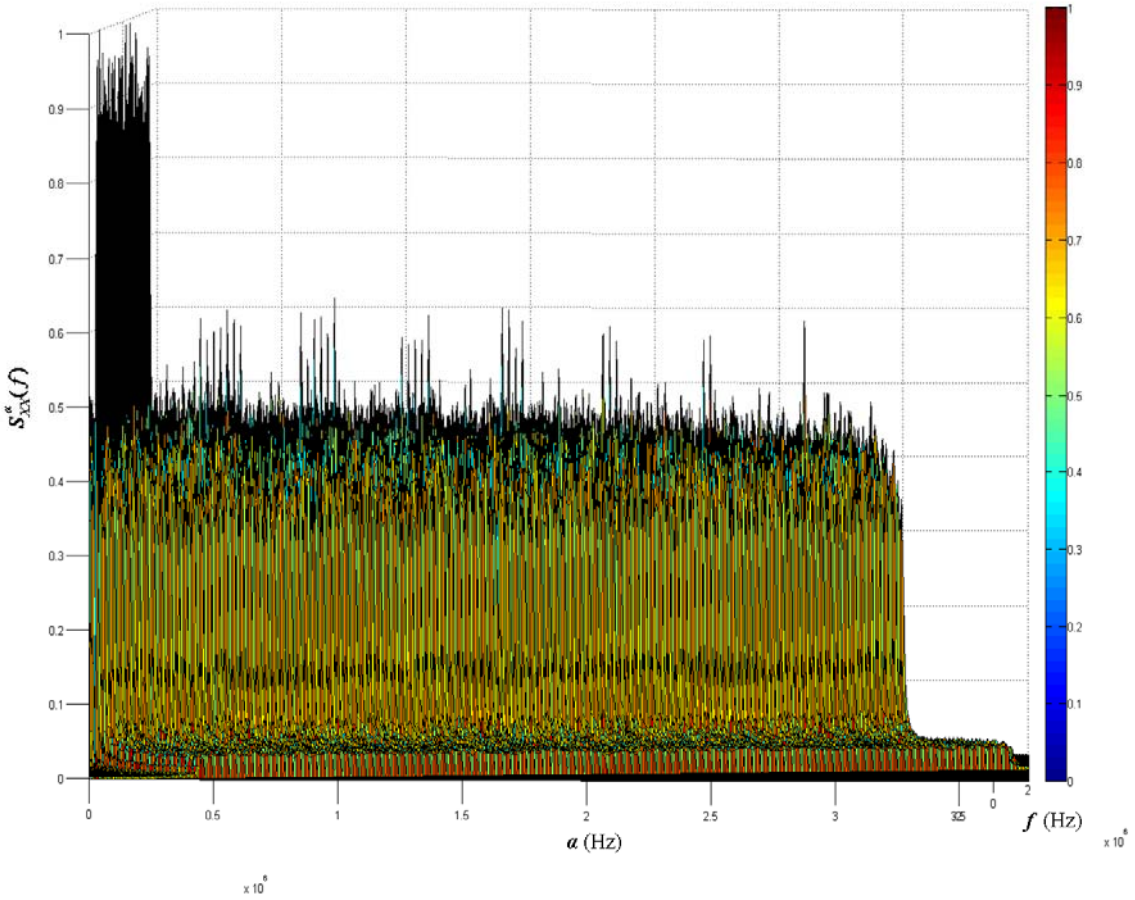

Figure 33. Surface Plot Representation of an IEEE 802.16 Waveform's $S^{\alpha}_{xx}(f)$. Results Averaged Over 120 Symbols.

Figure 34 shows the profile plots of the SCD magnitude as a function of the cyclic frequency α and frequency f. Figure 34(a) clearly shows all the pilot subcarrier peaks along the principle cyclic axis.

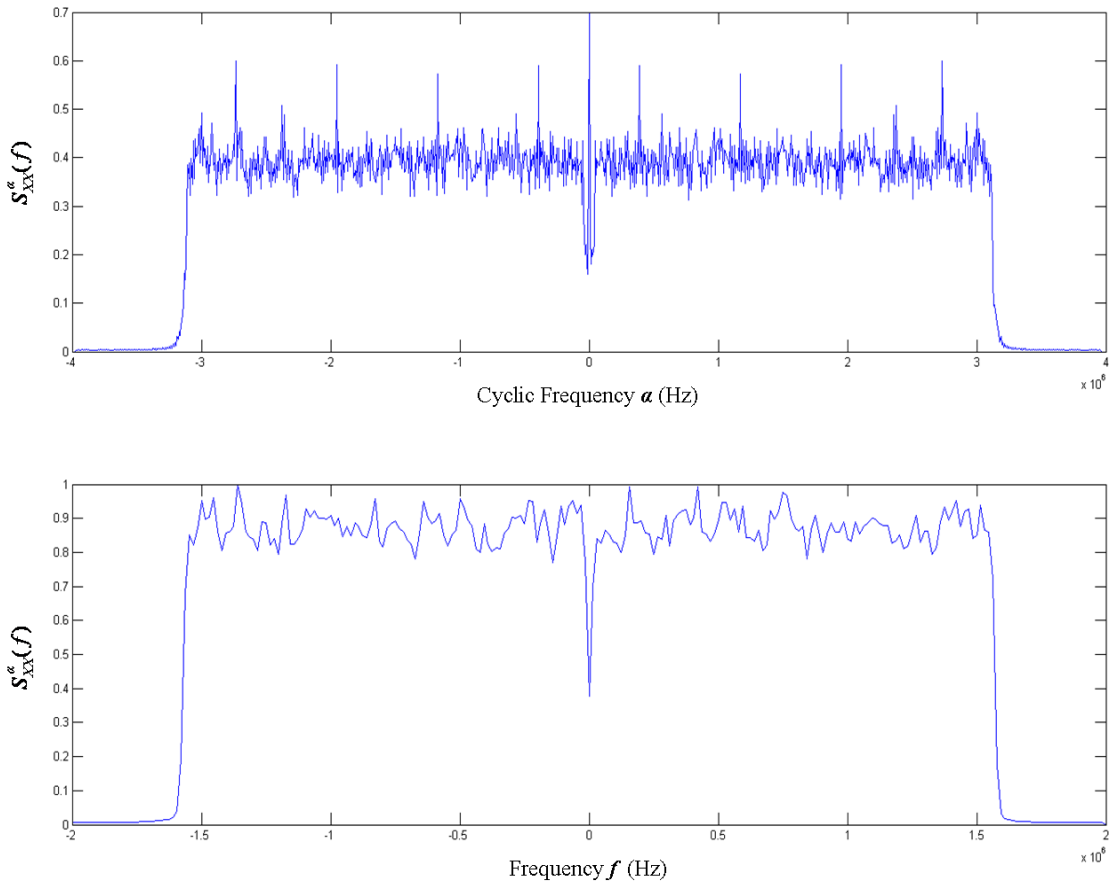

Figure 34. Profile Plots: Magnitude of $S_{xx}^{\alpha}(f)$ versus a) Cyclic Principal Frequency and b) Frequency for IEEE 802.16 Waveform. Results Averaged Over 120 Symbols.

Figure 35 provides the top down perspective of the pilot subcarrier cyclic peak locations. The contour level in Figure 34 was set at 0.54 to ensure all the pilot subcarrier cyclic spectral lines were represented, while excluding spurious responses due to random data on data subcarriers.

.

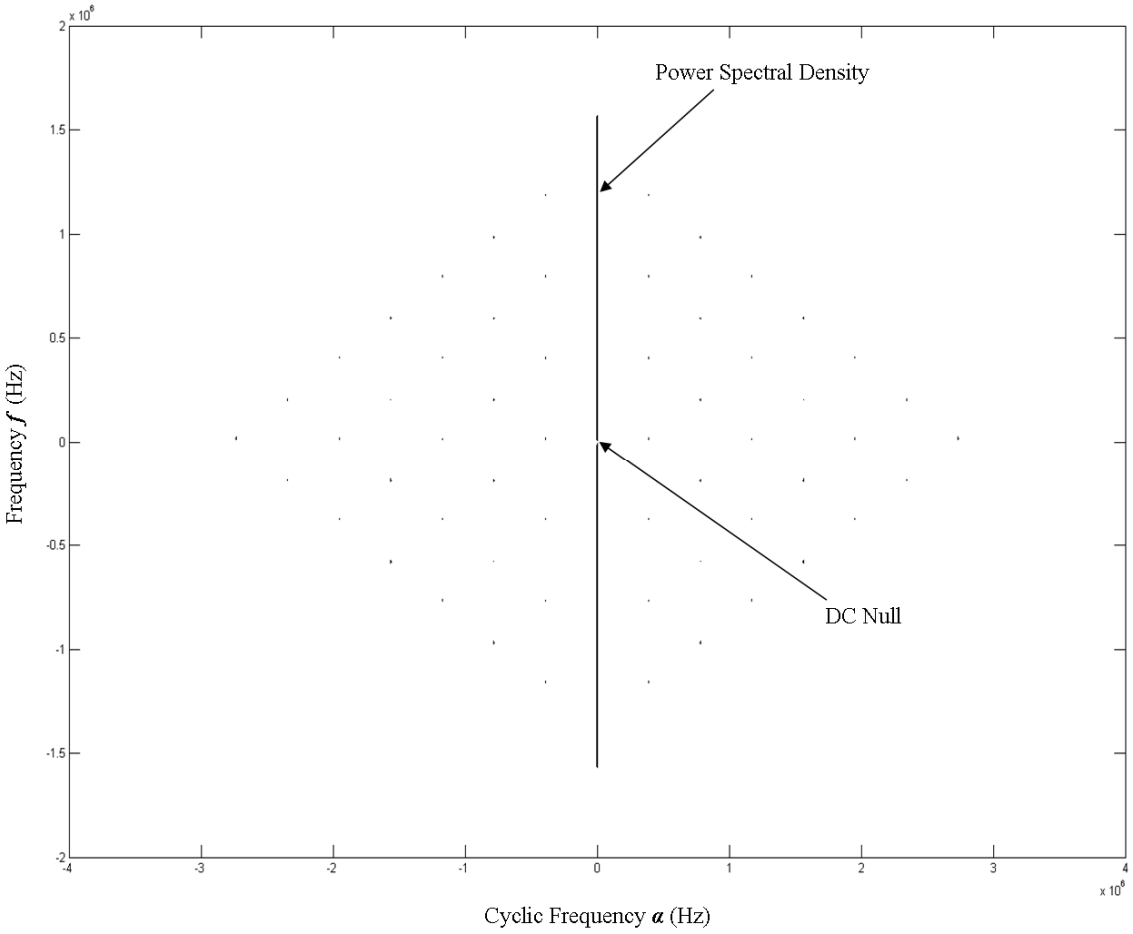

Figure 35. Contour Plot of an IEEE 802.16 Waveform's $S_{xx}^{\alpha}(f)$ Magnitude. Results Averaged Over 120 Symbols.

Results of Figures 36 and 37 are similar to those in Figure 31 and 32, respectively. Figure 36 demonstrated a clearly identifiable separation at -10 dB between the pilot and data subcarriers. The results displayed in Figure 37 illustrate how the cyclostationary signature of the pilot subcarriers is not identifiable until more than three symbols have been averaged. In order to satisfy the above mentioned classification criteria, roughly 10 OFDM symbols need to be averaged. Beyond 20 symbols, not much improvement was obtained.

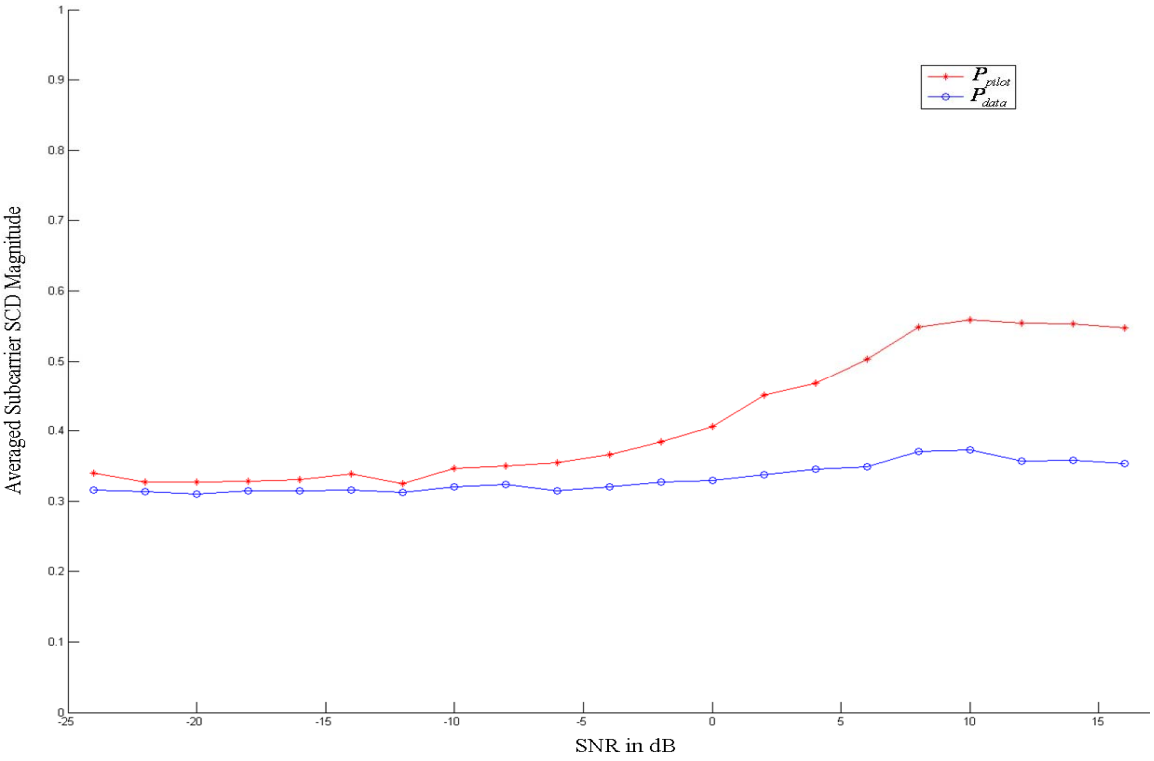

Figure 36. IEEE 802.16 Averaged Subcarrier Peak Values versus SNR.

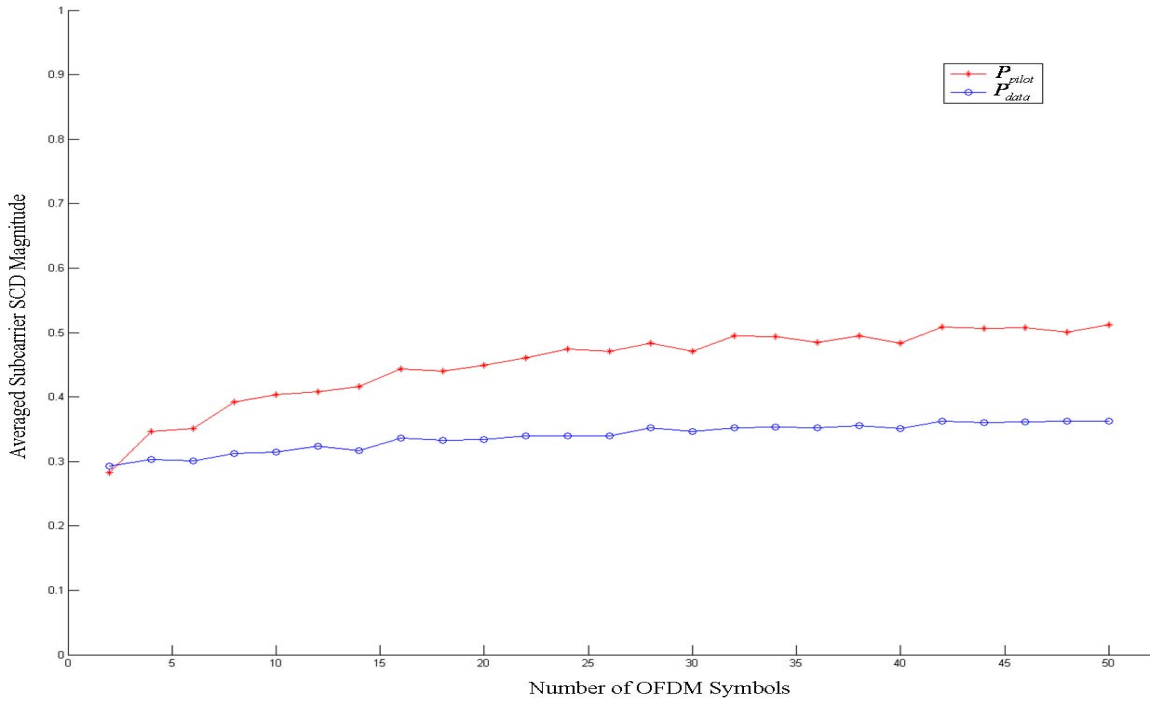

Figure 37. IEEE 802.16 Averaged Subcarrier Peak Values versus Number of OFDM Symbols Processed by the FAM SCD Function Estimator.

In summary, the surface, profile and contour plots of the previous section are clear illustrations of the effectiveness of the cyclostationary feature extraction techniques in regard to identifying and classifying OFDM based waveforms. Although the plots were generated by averaging 120 OFDM symbols, the implementation model coding is able to reliably identify the pilot subcarrier spectral peaks in as few as 20 OFDM symbols. In both cases, after 20 OFDM symbols are averaged, the pilot subcarriers always possessed a higher magnitude than the data carriers.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

This thesis discussed the fundamentals of OFDM and how OFDM modulation is implemented in the IEEE 802.11 and 802.16 standards. Cyclostationarity was then introduced to provide the reader with a basic understanding required for the following chapters. Next, a scheme for identifying and classifying 802.11 and 802.16 waveforms was proposed and discussed. Three possible cyclostationary feature extraction approaches were discussed and vetted for implementation. The most promising method, pilot subcarrier cyclostationary feature extraction, demonstrated the most potential and was chosen to classify the CP size of 802.16 waveforms. Additionally, pilot subcarrier cyclostationary feature extraction was employed to confirm the results of preamble cross-correlation identification process. Finally, a MATLAB simulation model was developed and the simulation results presented and discussed.

A. SIGNIFICANT RESULTS AND CONTRIBUTIONS

There were four major contributions made by this thesis. First, the work established an approach to identify a received waveform with no a priori knowledge of its origin. After much research and experimentation, it was determined that a preamble cross-correlation operation would provide the most reliable results while minimizing computational complexity of the simulations.

Next, a method of classifying the cyclic prefix length of IEEE 802.16 waveforms was devised. Since the CP option is established when the network is first setup, there is no way for a passive listener to identify this value by decoding captured control transmissions. By employing cyclostationary feature extraction, we were able to determine the CP length in a rapid fashion.

After identifying three methods of implementing cyclostationary feature extraction from [1], [3] and [4], we needed to determine which method or methods would be most compatible with the stated objectives of this work. Although embedded cyclostationary signature extraction is a promising technique, pilot subcarrier cyclostationary feature extraction proved to be the most applicable method. Unlike the

waveforms analyzed in [1], IEEE 802.11 and 802.16 waveforms did not amplify their pilot subcarrier transmissions so an averaging process had to be devised and implemented.

Finally, an overall scheme was developed to integrate the previous three processes into an organized framework for implementation.

B. FUTURE WORK

Initial identification of a received waveform was accomplished through a preamble correlation process. Although this method proved to be reliable, a rigorous application of detection theory was not applied due to time constraints. In order to generalize the results of the preamble correlation process, rigorous signal detection methods could be applied to this portion of the work to lend it greater credibility.

In this thesis, cyclostationary feature extraction was used to identify and classify OFDM based waveforms. The construction of the OFDM symbols in the two IEEE standards referenced in this thesis is a static process in that the format of the symbols remained the same. Other than in the preamble symbols, the null, pilot and data subcarrier locations remained the same. This is not the case with the IEEE 802.16e standard, which employs Orthogonal Frequency Division Multiple Access (OFDMA). OFDMA combines the baseband modulation operation and the multiple access process into one operation. Users are assigned a group of subcarriers, i.e., the number and location of data and pilot subcarriers will now vary with the number of users. So not only is the sequence that modulates the pilot subcarriers pseudo random, but with 802.16e, the location of many of the pilot subcarriers is also pseudo random, with respect to time. In order to obtain the pilot subcarrier cyclostationary signature, the Down Link MAP (DL-MAP) messages must be decoded to determine the number and location of both the fixed and non-fixed pilot subcarriers.

APPENDIX

This Appendix contains the MATLAB code used in the implementation model. The main program is listed first with the functions called by this program listed in order of their utilization.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          Thesis OFDM Signal Generator                      %
%                             Written by Steven Schnur                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         Select Waveform to be Created                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WiFi=1;
WiMax=2;
Waveform_Type=WiMax;     % Enter WiMax Or WiFi to create waveform
SNR=18.1;                 % Signal to Noise Ratio of AWGN channel
if SNR <= 8             % Logic based on SNR to determine Baseband Mod
    m=1;
    else if ((SNR > 8) && (SNR <= 12))
        m=2 ;
    else if ((SNR > 12) && (SNR <=18))
            m=4 ;
        else if (SNR > 18)
                m=6;
            end
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                Input settings for WiFi Signal Generator                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if Waveform_Type == WiFi
L=64;                    % FFT Size/Number of Subcarriers
FFT=L;                   %
N=200;                   % Number of OFDM symbols to be transmitted
SYM=N;
Frame=200;                % Number of OFDM Symbols per Frame
CP=ceil(L/4);            % Size of Cyclic Prefix
G=1 ;                    % PILOT Subcarrier Gain
Npilot=4;                % Number of Pilot subcarriers
Ndata=48;                % Number of Data subcarriers
bk=randint(1,m*Ndata*N,[0,1]);  % Input Data string generator
```

```
[x]=WiFi_BasebandMod(L,N,m,Ndata,bk,CP,G,Frame);  % Calls WiFi tx funct
                                                  % feeds required data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            Input settings for WiMax Signal Generator                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else if Waveform_Type == WiMax
L=256;                    % FFT Size/Number of Subcarriers
FFT=L;                    %
N=67 ;                   % Number of OFDM symbols to be transmitted 138
SYM=N;
Frame=138;                % Number of OFDM Symbols per Frame 138
CP=ceil(L/4);             % Size of Cyclic Prefix
G=1 ;                    % PILOT Subcarrier Gain
Npilot=8;                 % Number of Pilot subcarriers
Ndata=192;                % Number of Data subcarriers
bk=randint(1,m*Ndata*N,[0,1]);  % Input Data string generator
[x]=WiMax_BasebandMod(L,N,m,Ndata,bk,CP,G,Frame);% Calls WiMax tx funct
                                                  % andfeeds required
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              AWGN Channel                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y = awgn(x,SNR,'measured') ;  % Additive White Gaussian Channel

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Preamble Correlation                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[WiMax_Pre,WiFi_Pre]=Comp_Pre;            % Funt with sample Preample
CPsym=y(1:320,1);                         % Vect with first 320 samples
[Preamble_corr,lags] = xcorr(WiFi_Pre,CPsym,'biased');
[Preamble_corr2,lags2] = xcorr(WiMax_Pre',CPsym,'biased');
[WiFi_Peak,WiFi_Peak_Pos] = max(abs(Preamble_corr));
[WiMax_Peak,WiMAx_Peak_Pos] = max(abs(Preamble_corr2));
WiFi_Norm=abs(Preamble_corr);
WiMax_Norm=abs(Preamble_corr2);
xaxis=linspace(-319,319,639);
hold on
figure(7)
subplot(2,1,1), plot(xaxis,WiFi_Norm)

xlabel('Samples Shift')
ylabel('Normalized Preamble Correlation')
axis([-330 330 0 .015])
subplot(2,1,2), plot(xaxis,WiMax_Norm)
title('b) 802.16 Preamble/Recived Signal Cross Correlation')
xlabel('Sample Shift')

ylabel('Normalized Preamble Correlation')
axis([-330 330 0 .015])
hold off
```

```
if WiFi_Peak > .009 & WiMax_Peak > .009
   Signal_Type=num2str('System Error, Both Waveforms Identified')
else if WiFi_Peak < 0.009 & WiMax_Peak < 0.009
     Signal_Type=num2str('Unknown Waveform')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Input settings for WiFi FAM Spectral Correlation Density Function %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if WiFi_Peak > 0.009
fs=20000000;          % Sampling Freq used by FAM Estimate
df=ceil(20*10^6/128);     % Freq Resolution of FAM
dalpha=ceil(20*10^6/256); % Cyclic Freq Resolution of FAM
ss=80;              % Size of OFDM symbol with CP
index=44;          % Sets increment size of OFDM symbols for FAM
analysis 44
shift=256;          % Number of samples loaded into FAM Estimator
v=[.55];               % Height of contour plot plane

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              Call FAM function for Cyclostationary analysis           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

famin=y';                                % Transposes tx signal+AWGN vector
[Sxp,fo,Np,centrow,centcol]=WiFi_autofam(famin,fs,df,dalpha,ss,index,v,
...
    shift);

PilotSC_ave=sum([Sxp(centrow,89),Sxp(centrow,201),...
    Sxp(centrow,313),Sxp(centrow,425)])/4
DataSC_ave=sum([Sxp(centrow,49:88),Sxp(centrow,97:200),...
    Sxp(centrow,209:248),Sxp(centrow,265:312),Sxp(centrow,321:424),...
    Sxp(centrow,433:464)])/376
if PilotSC_ave > DataSC_ave*1.1
    Signal_Type=num2str('IEEE 802.11 Waveform Classified')
else
    Signal_Type=num2str('IEEE 802.11 Waveform Not Classified')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                WiFi Receiver Section                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ck]=WiFi_BasebandDemod(L,N,m,Ndata,CP,y,bk,Frame); % Calls rec funct
                                                    % feeds required
bk1=bk(1,1:length(ck));         % Adjusts length of input data for
err=bk1(1,:)~=ck(1,:);          % Compares rec data string to trans
BER=max(cumsum(err))/(m*Ndata*N)  % Calculates Bit Error Rate

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Input settings for WiMax FAM Spectral Correlation Density Function%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if WiMax_Peak > 0.009
fs=4000000;              % Sampling Freq used by FAM Estimate
df=ceil(15625);          % Freq Resolution of FAM
```

```
dalpha=ceil(15625/2);    % Cyclic Freq Res of FAM
ss=CP+L;                 % Size of OFDM Sym w/CP. Used to prep data for
FAM
index=30;                % Sets increment size of OFDM sym for FAM
analysis 68
shift=512;               % Num of data samples loaded into FAM each
iter
v=[.55];                 % Height of contour plot plane

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Code to Determine CP length                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

famin=y';
for num=1:4
    CP1=2^(num+2);      % Sets CP length for famtest
[Sxp,fo,Np,centrow,centcol]=WiMax_autofamtest(famin,fs,df,...
    dalpha,ss,index,CP1,shift);   % Generates SCD for compison
peak1(1,1)=(Sxp(centrow,163));
peak1(1,2)=(Sxp(centrow,263));
peak1(1,3)=(Sxp(centrow,363));
peak1(1,4)=(Sxp(centrow,463));
peak1(1,5)=(Sxp(centrow,563));
peak1(1,6)=(Sxp(centrow,663));
peak1(1,7)=(Sxp(centrow,763));
peak1(1,8)=(Sxp(centrow,863));
peak(1,num)=mean(peak1);                    % Determines mean of peak1
end                                            % vector

[Sigtype,location]=max(peak);
if location==1
        CP2=256/32
    Signal_Type=num2str('256 FFT with a 1/32 Cyclic Prefix')
else if location==2
        CP2=256/16
        Signal_Type=num2str('256 FFT with a 1/16 Cyclic Prefix')
    else if location==3
            CP2=256/8
            Signal_Type=num2str('256 FFT with a 1/8 Cyclic Prefix')
        else if location==4
                CP2=256/4
                Signal_Type=num2str('256 FFT with a 1/4 Cyclic Prefix')
            end
        end
    end
end

[Sxp,fo,Np,centrow,centcol]=WiMax_autofam(famin,fs,df,dalpha,ss,index,v
,...
    shift,CP2);     % Creates SCD and graphs for the CP signal

PilotSC_ave=sum([Sxp(centrow,163),Sxp(centrow,263),...
    Sxp(centrow,363),Sxp(centrow,463),Sxp(centrow,563),...
    Sxp(centrow,663),Sxp(centrow,763),Sxp(centrow,863)])/8
```

```
DataSC_ave=sum([Sxp(centrow,117:160),Sxp(centrow,165:260),...
    Sxp(centrow,265:360),Sxp(centrow,365:460),Sxp(centrow,465:512),...
    Sxp(centrow,517:560),Sxp(centrow,565:660),Sxp(centrow,665:760),...
    Sxp(centrow,765:860),Sxp(centrow,865:913)])/761
if PilotSC_ave > DataSC_ave*(1.1)
    Signal_Type=num2str('IEEE 802.16 Waveform Classified')
else
    Signal_Type=num2str('IEEE 802.16 Waveform Not Classified')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        WiMax Receiver Section                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[ck]=   WiMax_BasebandDemod(L,N,m,Ndata,CP,y,bk,Frame);%Calls rec funct
                                                % and feeds required data
bk1=bk(1,1:length(ck));      % Adjusts length of input data for compare
err=bk1(1,:)~=ck(1,:);          % Compares rec data string to trans
BER=max(cumsum(err))/(m*Ndata*N)  % Calculates Bit Error Rate end

end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        Baseband Data Setup and Modulator                %
%                           Written by Steven Schnur                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x]=WiFi_BasebandMod(L,N,m,Ndata,bk,CP,G,Frame);
if nargin~=8
    error('Wrong number of arguments')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Generates 64QAM Modulation Object                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

if m==6
c=sqrt(42);                                 % Normalization Value of 64QAM
Sym
object = modem.genqammod('Constellation', [ (-7-7j)/c,(7-7j)/c,...
    (-1-7j)/c, (1-7j)/c, (-5-7j)/c, (5-7j)/c, (-3-7j)/c, (3-7j)/c, ...
    (-7+7j)/c, (+7+7j)/c, (-1+7j)/c,(1+7j)/c, (-5+7j)/c, (5+7j)/c,...
    (-3+7j)/c, (3+7j)/c, (-7-1j)/c, (7-1j)/c, (-1-1j)/c,(1-1j)/c,...
    (-5-1j)/c, (5-1j)/c, (-3-1j)/c, (3-1j)/c  (-7+1j)/c,(7+1j)/c,...
    (-1+1j)/c,(1+1j)/c, (-5+1j)/c, (5+1j)/c, (-3+1j)/c, (3+1j)/c,...
    (-7-5j)/c, (+7-5j)/c, (-1-5j)/c  (1-5j)/c, (-5-5j)/c, (5-5j)/c,...
    (-3-5j)/c, (3-5j)/c, (-7+5j)/c, (7+5j)/c, (-1+5j)/c, (1+5j)/c,...
    (-5+5j)/c, (5+5j)/c, (-3+5j)/c, (3+5j)/c, (-7-3j)/c, (7-3j)/c,...
    (-1-3j)/c, (1-3j)/c, (-5-3j)/c, (5-3j)/c, (-3-3j)/c, (3-3j)/c,...
    (-7+3j)/c, (7+3j)/c, (-1+3j)/c, (1+3j)/c, (-5+3j)/c, (5+3j)/c,...
```

```
    (-3+3j)/c, (+3+3j)/c ], ...
    'InputType', 'Bit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Generates 16QAM Modulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else if m==4
c=sqrt(10);                                % Normalization Value of 16QAM
Sym
object = modem.genqammod('Constellation', [ (-3-3j)./c, ...
     (3-3j)./c, (-1-3j)./c, (1-3j)./c, (-3+3j)./c, (3+3j)./c,...
     (-1+3j)./c, (1+3j)./c, (-3-1j)./c, (3-1j)./c, (-1-1j)./c,...
     (1-1j)./c, (-3+1j)./c, (3+1j)./c,(-1+1j)./c, (1+1j)./c ], ...
    'InputType', 'Bit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Generates QPSK Modulation Object                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==2
c=sqrt(2);
object = modem.genqammod('Constellation', [ (-1-1j)/c,  (1-1j)/c, ...
    (-1+1j)/c, (1+1j)/c ], ...
    'InputType', 'Bit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Generates BPSK Modulation Object                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==1
object = modem.genqammod('Constellation', [ -1, 1 ], ...
        'InputType', 'Bit');
    end                        % End of BPSK mod loop
    end                        % End of QPSK mod loop
    end                        % End of 16QAM mod loop
end                            % End of 64QAM mod loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                Baseband Modulation                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

temp=bk;                                           % Dummy Vectors to load
pad=0;                                             %   preamble
index=mod([-3:N],Frame) & mod([-2:N+1],Frame)...
    & mod([-1:N+2],Frame) & mod([0:N+3],Frame);%Creats4logic 4 preamble
                                                   %   insertion

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Pilot Subcarrier Sequence Generator                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pilot_SC=[ 1,1,1,1, -1,-1,-1,1, -1,-1,-1,-1, 1,1,-1,1, -1,-1,1,1,...
    -1,1,1,-1, 1,1,1,1, 1,1,-1,1, 1,1,-1,1, 1,-1,-1,1, 1,1,-1,1, ...
    -1,-1,-1,1, -1,1,-1,-1, 1,-1,-1,1, 1,1,1,1, -1,-1,1,1,...
    -1,-1,1,-1, 1,-1,1,1, -1,-1,-1,1, 1,-1,-1,-1, -1,1,-1,-1, ...
```

```
    1,-1,1,1, 1,1,-1,1, -1,1,-1,1, -1,-1,-1,-1, -1,1,-1,1, 1,-1,1,-1,
...
    1,1,1,-1, -1,1,-1,-1, -1,1,1,1, -1,-1,-1,-1, -1,-1,-1 ];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  Outer loop to form overall tx data vector                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:N                                  % Loop to generate OFDM Symbols

     if ((index(1,k)==0) & (index(1,k+3)==0));% Logic operation to load
        [short1,short2]=WiFi_short_preamble;
        x1=short1;

        else if (index(1,k)==0)&(index(1,k+2)==0)&(index(1,k+3)==1);
                                                 % Logic op to load
        [short1,short2]=WiFi_short_preamble;
        x1=short2;          %
        else if (index(1,k)==0)&(index(1,k+1)==0)&(index(1,k+2)==1);
        [long1,long2]=WiFi_long_preamble;
        x1=long1;
        else if (index(1,k)==0)&(index(1,k+1)==1);
        [long1,long2]=WiFi_long_preamble;
        x1=long2;
        bkp=wextend('addcol','zpd',temp,Ndata*m*4,'r'); % Makes room in
        bkp=circshift(bkp,[0,Ndata*m*4]);          %   data vector for
        temp=bkp;
    else if index(1,k)==1;                      % Loads data if preamble
    x1=0;                                        %   not met
    a=bkp(1,Ndata*m*(k-1)+1:m*Ndata*k);     % Loads one symbol worth of
    xt=0;                                        %   data in a storage vec

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Inner loop to form OFDM symbol                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for w=1:Ndata                    % Loop to load m-by-Ndata fft vector
    aa=a(1,(m)*(w-1)+1:m*w);     % Grabs one m-symbol worth of data
    mk=aa(1,1:m);                % Loading of storage vect with 1 m-sym
    Xp=modulate(object,mk');   % Modulates m sized sym with obj: IQ data
    X1(w,1)=Xp;                  % Stacks IQ data in a column vector
end                              % End of w=1:Ndata loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load Vector X with Guard SC=0, DC=0, Pilot SC and Data in                  %
%                      appropriate SC locations                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X(1,1)=0;                              % DC Null
X(2:7,1)=X1(25:30,1);                  % Data
X(8,1)=Pilot_SC(1,2)*G*1;                % Positive Pilot SC
X(9:21,1)=X1(31:43,1);                 % Data
X(22,1)=Pilot_SC(1,2)*G*-1;              % Positive Pilot SC
X(23:27,1)=X1(44:48,1);                % Data
```

```
X(28:32,1)=0;                       % Lower Guard Band
X(33:38,1)=0;                       % Upper Guard Band
X(39:43,1)=X1(1:5,1);               % Data
X(44,1)=Pilot_SC(1,2)*G*1;            % Negitive Pilot SC
X(45:57,1)=X1(6:18,1);              % Data
X(58,1)=Pilot_SC(1,2)*G*1;            % Negitive Pilot SC
X(59:64,1)=X1(19:24,1);             % Data
Pilot_SC=circshift(Pilot_SC,[0,-1]);% Pseudo-randomize pilot seq IAW
Xplot((k-1)*L+1:k*L,1)=X;           %   WiFi standard
        end             % End of else if index(1,k)==1 loop


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           IFFT Baseband Modulated data to generate time data        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xt=ifft(X,L);       % L size IFFT to convert I-Q data to time samples
g(1:CP,1)=xt((L-CP+1):L,1); % Loads CP worth of Data end of OFDM sym
x1(1:CP,1)=g;               % Loads CP to front of vector
x1(CP+1:L+CP,1)=xt;         % Loads rest of OFDM Symbol into vector
end                         % End line 97 loop
end                         % End line 95 loop
end                         % End line 93 loop
end                         % End line 90 loop
x(((k-1)*(L+CP)+1):k*(L+CP),1)=x1; % Fills tx vector w/CP appended
end                         % End for k=1:N loop
figure(1)
plot(X,'*');                % Plot Transmitted Constellation
title('Transmitted I-Q Data')
xlabel('Transmitted In Phase Data')
ylabel('Transmitted Quadrature Data')
axis([-1.5 1.5 -1.5 1.5]);
grid on


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     WiFi Short Preamble Time Samples                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [short1,short2]=WiFi_short_preamble;


short1=[0.023+0.023j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
```

```
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j];

short2=[0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002+0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002+0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013+0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j];



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        WiFi Long Preamble Time Samples
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [long1,long2]=WiFi_long_preamble;
long1=[-0.078+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003+0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j;...
0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j;...
0.156+0.000j;-0.005-0.120j;0.040-0.111j;0.097+0.083j;...
0.021+0.028j;0.060-0.088j;-0.115-0.055j;-0.038-0.106j;...
0.098-0.026j;0.053+0.004j;0.001-0.115j;-0.137+0.047j;...
0.024-0.059j;0.059-0.015j;-0.022+0.161j;0.119-0.004j;...
0.062-0.062j;0.037+0.098j;-0.057+0.039j;-0.131+0.065;...
0.082+0.092j;0.070+0.014j;-0.060+0.081j;-0.056-0.022j;...
-0.035-0.151j;-0.122-0.017j;-0.127-0.021j;0.075-0.074j;...
-0.003+0.054j;-0.092+0.115j;0.092+0.106j;0.012+0.098j;...
-0.156+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003-0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
```

```
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j];

long2=[0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j;...
0.156+0.000j;-0.005-0.120j;0.040-0.111j;0.097+0.083j;...
0.021+0.028j;0.060-0.088j;-0.115-0.055j;-0.038-0.106j;...
0.098-0.026j;0.053+0.004j;0.001-0.115j;-0.137-0.047j;...
0.024-0.059j;0.059-0.015j;-0.022+0.161j;0.119-0.004j;...
0.062-0.062j;0.037+0.098j;-0.057+0.039j;-0.131+0.065j;...
0.082+0.092j;0.070+0.014j;-0.060+0.081j;-0.056-0.022j;...
-0.035-0.151j;-0.122-0.017j;-0.127-0.021j;0.075-0.074j;...
-0.003+0.054j;-0.092+0.115j;0.092+0.106j;0.012+0.098j;...
-0.156+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003-0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j;...
0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Reference Waveform Samples For Preamble Cross-Correlation Process %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [WiMax_Pre,WiFi_Pre]=Comp_Pre;
WiMax_Pre=1*...
[-0.0,0.0997 - 0.0305i,-0.0055 + 0.0140i,0.0949 + 0.1927i,...
0.0735 + 0.0892i,-0.0145 - 0.0140i,0.1582 + 0.1195i,-0.0099 -
0.0046i...
-0.1943 + 0.0169i,-0.0331 + 0.0719i,-0.0051 - 0.0253i,0.0860 +
0.1501i,...
0.0135 + 0.0133i,-0.3556 - 0.1717i,-0.1981 - 0.0080i,0.0422 -
0.0174i,...
-0.1171 + 0.1094i,-0.0243 + 0.0399i,0.0955 - 0.2904i,0.0060 -
0.1276i,...
0.0074 + 0.0105i,-0.0298 + 0.0183i,-0.0173 + 0.0291i,0.0652 -
0.1788i,...
0.0298 - 0.1207i,-0.0030 + 0.0140i,0.0056 - 0.0015i,0.0111 +
0.0225i,...
-0.0780 - 0.1439i,-0.1135 - 0.1640i,0.0244 + 0.0283i,-0.0426 -
0.0429i,...
-0.0690 - 0.0690i,0.1682 + 0.1992i,0.1395 + 0.2335i,-0.0107 -
0.0293i,...
0.0069 + 0.0325i,0.0148 + 0.0936i,-0.0417 - 0.1614i,-0.0919 -
0.1099i,...
0.0096 - 0.0003i,-0.0060 + 0.0033i,-0.0985 + 0.0748i,0.2024 -
0.1640i,...
0.3047 - 0.2307i,-0.0023 + 0.0014i,-0.0111 + 0.0044i,0.0511 -
0.0051i,...
-0.0375 - 0.0104i,0.0088 + 0.0020i,0.0043 + 0.0039i,-0.0177 -
0.0206i,...
```

```
0.0699 + 0.0884i,-0.1185 - 0.1416i,-0.3045 - 0.2972i,-0.0372 -
0.0232i,...
0.0511 + 0.0064i,-0.0678 + 0.0447i,0.0417 - 0.1090i,-0.0145 -
0.4059i,...
-0.0286 - 0.0767i,0.0594 + 0.0694i,-0.0872 - 0.0368i,0.0862 -
0.0173i,...
0.2500 - 0.2500i,0.0322 - 0.0993i,0.0091 + 0.1087i,-0.0406 -
0.0795i,...
0.0542 + 0.0439i,0.3314 + 0.0793i,0.1083 - 0.0225i,-0.0699 +
0.0492i,...
0.0226 - 0.0352i,-0.0035 + 0.0209i,0.0373 + 0.1116i,0.0257 +
0.0130i,...
-0.0104 + 0.0136i,-0.0033 - 0.0094i,0.0052 + 0.0040i,0.1601 +
0.0383i,...
0.1071 - 0.0129i,-0.0704 + 0.0317i,0.0095 - 0.0075i,0.0014 -
0.0015i,...
-0.1627 + 0.1990i,-0.1191 + 0.1293i,0.0786 - 0.0642i,0.0073 -
0.0043i,...
-0.0194 + 0.0087i,0.3171 - 0.1223i,0.2274 - 0.0910i,-0.0751 +
0.0409i,...
-0.0111 + 0.0141i,-0.0036 - 0.0150i,0.1417 + 0.1583i,0.2027 +
0.1832i,...
-0.0819 - 0.0819i,-0.0446 - 0.0605i,0.0189 + 0.0445i,-0.0329 -
0.2830i,...
0.0508 - 0.2856i,-0.0300 + 0.0631i,-0.0340 + 0.0385i,0.0112 -
0.0030i,...
0.0015 - 0.0661i,0.0906 - 0.1662i,-0.0337 + 0.0411i,-0.0725 +
0.0656i,...
0.0271 - 0.0165i,-0.0871 + 0.0116i,-0.0527 - 0.0869i,-0.0046 +
0.0219i,...
-0.0443 + 0.0919i,0.0172 - 0.0271i,-0.0717 + 0.0985i,-0.0956 +
0.1266i,...
0.0018 - 0.0032i,-0.0076 - 0.0156i,0.0044 + 0.0034i,-0.0200 -
0.0345i,...
-0.0086 - 0.1510i,0.0021 - 0.0082i,-0.0705 + 0.1481i,0.0110 -
0.0159i,...
-0.0990 + 0.1029i,-0.3336 + 0.2360i,-0.0441 + 0.0178i,0.2044 -
0.0258i,...
0.0000,0.1005 + 0.0285i,0.3725 + 0.1423i,0.0798 + 0.0336i,...
-0.2165 - 0.0945i,-0.0218 - 0.0103i,-0.0670 - 0.0381i,-0.2929 -
0.2231i,...
-0.0748 - 0.0822i,0.1079 + 0.1809i,0.0141 + 0.0397i,0.0057 +
0.0368i,...
-0.0129 + 0.1843i,-0.0204 + 0.0303i,0.0395 + 0.0382i,0.0109 +
0.0408i,...
0.0013 + 0.0248i,-0.0377 + 0.3309i,-0.0533 + 0.1767i,0.1048 -
0.1816i,...
0.0641 - 0.0578i,0.0092 - 0.0031i,0.1928 + 0.0235i,0.0948 + 0.0611i,...
-0.0407 - 0.0806i,0.0165 - 0.0461i,0.0010 - 0.0005i,0.2000 +
0.0022i,...
0.1978 + 0.0482i,-0.1638 - 0.0659i,-0.1127 - 0.0597i,0.0018 +
0.0012i,...
-0.0248 - 0.0248i,0.0599 + 0.0358i,-0.0880 - 0.0643i,-0.1278 -
0.1008i,...
```

```
0.0041 + 0.0033i,-0.1568 - 0.1162i,-0.1636 - 0.0907i,0.0742 +
0.0122i,...
0.0886 - 0.0377i,-0.0008 + 0.0008i,0.0790 - 0.0893i,0.1306 -
0.1156i,...
-0.0626 + 0.0264i,-0.1228 - 0.0187i,-0.0029 - 0.0024i,-0.0642 -
0.1187i,...
-0.0783 - 0.2822i,0.0120 + 0.1022i,-0.0010 + 0.2725i,-0.0020 +
0.0175i,...
-0.0257 + 0.1027i,-0.0794 + 0.1700i,0.0223 - 0.0248i,0.1218 -
0.0640i,...
0.0198 - 0.0040i,0.0720 - 0.0004i,0.2306 + 0.0377i,-0.0258 -
0.0097i,...
-0.2046 - 0.1453i,-0.0303 - 0.0412i,-0.0242 - 0.0758i,-0.0144 -
0.3266i,...
0 + 0.0000i,-0.0803 + 0.3119i,-0.0190 + 0.0673i,-0.0037 + 0.0318i,...
0.0604 + 0.0829i,0.0173 + 0.0023i,-0.2430 + 0.0084i,-0.1050 -
0.0014i,...
-0.0369 - 0.0062i,-0.2672 - 0.1095i,-0.0531 - 0.0394i,0.1920 +
0.2163i,...
0.0670 + 0.0915i,0.0081 + 0.0089i,0.1187 + 0.0357i,0.0513 - 0.0255i,...
-0.1787 + 0.1912i,-0.0991 + 0.1499i,-0.0029 + 0.0060i,-0.0600 +
0.2146i,...
0.0091 + 0.0784i,-0.1196 - 0.0696i,-0.1351 + 0.0341i,0.0018 -
0.0012i,...
-0.1648 + 0.1744i,-0.0797 + 0.1429i,0.0440 - 0.2606i,-0.0547 -
0.1949i,...
0.0032 + 0.0040i,-0.1447 - 0.1054i,-0.1309 - 0.0716i,0.1392 +
0.0745i,...
0.0507 + 0.0507i,0.0008 - 0.0005i,-0.0924 + 0.0075i,-0.1597 -
0.0055i,...
0.2167 + 0.0247i,0.2508 + 0.0523i,0.0017 + 0.0006i,0.0670 + 0.0444i,...
0.0588 + 0.0858i,-0.0197 - 0.1036i,0.0143 - 0.1858i,0.0021 -
0.0096i,...
0.0272 - 0.0820i,0.0851 - 0.1811i,-0.0987 + 0.1418i,-0.2467 +
0.2218i,...
-0.0274 + 0.0132i,-0.0791 + 0.0117i,-0.2425 - 0.0328i,0.1475 +
0.0582i,...
0.2861 + 0.1857i,0.0276 + 0.0251i,0.0131 + 0.0151i,0.0111 + 0.0001i,...
0.0077 + 0.0420i,0.0180 + 0.2580i,-0.0074 + 0.0690i,-0.0077 +
0.0225i,...
-0.1333 + 0.1966i,0.0670 - 0.0549i,0.3598 - 0.1604i,0.0952 -
0.0216i,...
-0.0000,0.0997 - 0.0305i,-0.0055 + 0.0140i,0.0949 + 0.1927i,...
0.0735 + 0.0892i,-0.0145 - 0.0140i,0.1582 + 0.1195i,-0.0099 -
0.0046i,...
-0.1943 + 0.0169i,-0.0331 + 0.0719i,-0.0051 - 0.0253i,0.0860 +
0.1501i,...
0.0135 + 0.0133i,-0.3556 - 0.1717i,-0.1981 - 0.0080i,0.0422 -
0.0174i,...
-0.1171 + 0.1094i,-0.0243 + 0.0399i,0.0955 - 0.2904i,0.0060 -
0.1276i,...
0.0074 + 0.0105i,-0.0298 + 0.0183i,-0.0173 + 0.0291i,0.0652 -
0.1788i,...
0.0298 - 0.1207i,-0.0030 + 0.0140i,0.0056 - 0.0015i,0.0111 +
0.0225i,...
```

```
-0.0780 - 0.1439i,-0.1135 - 0.1640i,0.0244 + 0.0283i,-0.0426 -
0.0429i,...
-0.0690 - 0.0690i,0.1682 + 0.1992i,0.1395 + 0.2335i,-0.0107 -
0.0293i,...
0.0069 + 0.0325i,0.0148 + 0.0936i,-0.0417 - 0.1614i,-0.0919 -
0.1099i,...
0.0096 - 0.0003i,-0.0060 + 0.0033i,-0.0985 + 0.0748i,0.2024 -
0.1640i,...
0.3047 - 0.2307i,-0.0023 + 0.0014i,-0.0111 + 0.0044i,0.0511 -
0.0051i,...
-0.0375 - 0.0104i,0.0088 + 0.0020i,0.0043 + 0.0039i,-0.0177 -
0.0206i,...
0.0699 + 0.0884i,-0.1185 - 0.1416i,-0.3045 - 0.2972i,-0.0372 -
0.0232i,...
0.0511 + 0.0064i,-0.0678 + 0.0447i,0.0417 - 0.1090i,-0.0145 -
0.4059i,...
-0.0286 - 0.0767i,0.0594 + 0.0694i,-0.0872 - 0.0368i,0.0862 -
0.0173i]';
```

```
WiFi_Pre=[0.023+0.023j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002+0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013-0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
```

```
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002+0.132j;...
0.046+0.046j;-0.132+0.002j;-0.013+0.079j;0.143-0.013j;...
0.092+0.000j;0.143-0.013j;-0.013-0.079j;-0.132+0.002j;...
0.046+0.046j;0.002-0.132j;-0.079-0.013j;-0.013+0.143j;...
0.000+0.092j;-0.013+0.143j;-0.079-0.013j;0.002-0.132j;...
-0.078+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003+0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j;...
0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j;...
0.156+0.000j;-0.005-0.120j;0.040-0.111j;0.097+0.083j;...
0.021+0.028j;0.060-0.088j;-0.115-0.055j;-0.038-0.106j;...
0.098-0.026j;0.053+0.004j;0.001-0.115j;-0.137+0.047j;...
0.024-0.059j;0.059-0.015j;-0.022+0.161j;0.119-0.004j;...
0.062-0.062j;0.037+0.098j;-0.057+0.039j;-0.131+0.065;...
0.082+0.092j;0.070+0.014j;-0.060+0.081j;-0.056-0.022j;...
-0.035-0.151j;-0.122-0.017j;-0.127-0.021j;0.075-0.074j;...
-0.003+0.054j;-0.092+0.115j;0.092+0.106j;0.012+0.098j;...
-0.156+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003-0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j;...
0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j;...
0.156+0.000j;-0.005-0.120j;0.040-0.111j;0.097+0.083j;...
0.021+0.028j;0.060-0.088j;-0.115-0.055j;-0.038-0.106j;...
0.098-0.026j;0.053+0.004j;0.001-0.115j;-0.137-0.047j;...
0.024-0.059j;0.059-0.015j;-0.022+0.161j;0.119-0.004j;...
0.062-0.062j;0.037+0.098j;-0.057+0.039j;-0.131+0.065j;...
0.082+0.092j;0.070+0.014j;-0.060+0.081j;-0.056-0.022j;...
-0.035-0.151j;-0.122-0.017j;-0.127-0.021j;0.075-0.074j;...
-0.003+0.054j;-0.092+0.115j;0.092+0.106j;0.012+0.098j;...
-0.156+0.000j;0.012-0.098j;0.092-0.106j;-0.092-0.115j;...
-0.003-0.054j;0.075+0.074j;-0.127+0.021j;-0.122+0.017j;...
-0.035+0.151j;-0.056+0.022j;-0.060-0.081j;0.070-0.014j;...
0.082-0.092j;-0.131-0.065j;-0.057-0.039j;0.037-0.098j;...
0.062+0.062j;0.119+0.004j;-0.022-0.161j;0.059+0.015j;...
0.024+0.059j;-0.137+0.047j;0.001+0.115j;0.053-0.004j;...
0.098+0.026j;-0.038+0.106j;-0.115+0.055j;0.060+0.088j;...
0.021-0.028j;0.097-0.083j;0.040+0.111j;-0.005+0.120j;];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          WiFi Auto FAM (After [11])                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function
[Sxp,fo,Np,centrow,centcol]=WiFi_autofam(famin,fs,df,dalpha,ss,...
    index,v,shift);
```

```
if nargin~=8
    error('Wrong number of arguments.');
end
for ww=6:index*2
    xin(1,(ss-16)*(ww-1)+1:(ss-16)*ww)=famin(1,ss*(ww-1)+16+1:ss*ww);
end
%define parameters
Np=pow2(nextpow2(fs/df));
L=Np/4;
P=pow2(nextpow2(fs/dalpha/L));
N=P*L;
Sxp=zeros(Np+1,2*N+1);
for w=1:1:index/2
    x=xin(1,(shift)*(w-1)+1:(shift)*w);

   %input channalization
if length(x)<N
    x(N)=0;
elseif length(x)>N
    x=x(1:N);
end
NN=(P-1)*L+Np;
xx=x;
xx(NN)=0;
xx=xx(:);
X=zeros(Np,P);
for k=0:P-1
    X(:,k+1)=xx(k*L+1:k*L+Np);
end

%windowing

a=hamming(Np);
XW=diag(a)*X;
%XW=X;

%first FFT

XF1=fft(XW);
XF1=fftshift(XF1);
XF=[XF1(:,P/2+1:P) XF1(:,1:P/2)];

%downconvertions
E=zeros(Np,P);
for k=-Np/2:Np/2-1
    for k0=0:P-1
        E(k+Np/2+1,k0+1)=exp(-i*2*pi*k*k0*L/Np);
    end
end
XD=XF1.*E;
XD=conj(XD');

%multiplication
XM=zeros(P,Np^2);
```

```
for k=1:Np
    for c=1:Np
        XM(:,(k-1)*Np+c)=(XD(:,k).*conj(XD(:,c)));
    end
end

%second FFT
XF2=fft(XM);
XF2=fftshift(XF2);
XF2=[XF2(:,Np^2/2+1:Np^2) XF2(:,1:Np^2/2)];
XF2=XF2(P/4:3*P/4,:);
M=abs(XF2);
alphao=-fs:fs/N:fs;
fo=-fs/2:fs/Np:fs/2;
Sx=zeros(Np+1,2*N+1);
for k1=1:P/2+1
    for k2=1:Np^2
        if rem(k2,Np)==0
            c=Np/2-1;
        else
            c=rem(k2,Np)-Np/2-1;
        end
        k=ceil(k2/Np)-Np/2-1;
        p=k1-P/4-1;
        alpha=(k-c)/Np+(p-1)/L/P;
        f=(k+c)/2/Np;
        if alpha<-1 | alpha>1
            k2=k2+1;
        elseif f<-.5 | f>.5
            k2=k2+1;
        else
            kk=1+Np*(f+.5);
            ll=1+N*(alpha+1);
            Sx(round(kk),round(ll))=M(k1,k2);
        end
    end
end
Sxp=Sxp+Sx;
end
Sxp=Sxp./max(max(Sxp));

%    Surface Plot

Alpha=num2str(round(1+N*(alpha/fs+1)));
Fo=num2str(floor((Np+1)/2)+1);
V=num2str(v);
figure(4)
surfl(alphao,fo,Sxp);
view(-37.5,60);
xlabel('alpha');
ylabel('f');
zlabel('Sx');
colormap jet
colorbar
axis([0 20*10^6 -9*10^6 9*10^6 0 1])
```

```
%Contour Plot
figure(5)
contour(alphao,fo,Sxp,v);
xlabel('alpha,fo,Sxp');
ylabel('f(Hz)');
colormap winter
colorbar


%Cross-Section Plot
figure(6)
subplot(2,1,1), plot(alphao,Sxp(floor((Np+1)/2)+1,:));
xlabel('alpha(Hz)');
ylabel(['Sxp(',Fo,')'])
subplot(2,1,2), plot(fo,Sxp(:,round(1+N*(alpha/fs+1))));
xlabel('f(Hz)');
ylabel(['Sxp(',Alpha,')']);
centrow=floor((Np+1)/2)+1;
centcol=round(1+N*(alpha/fs+1));




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Baseband Data Setup and Demodulator                    %
%                         Written by Steven Schnur                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ck]=WiFi_BasebandDemod(L,N,m,Ndata,CP,y,bk,Frame);
if nargin~=8
    error('Wrong number of arguments')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              Remove Preamble                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

index=mod([-3:N],Frame) & mod([-2:N+1],Frame)...
    & mod([-1:N+2],Frame) & mod([0:N+3],Frame); %Creats logic 4preamble
                                                %   insertion
pad=0;
for b=1:N                   % Loop to break data string into L-fft sized
    Yp=0;                    %   blocks
    mpk=0;
    y1=0;

if (index(1,b)==0);          % Loop skips demod code if preamble logic
                             %   satisfied
else if (index(1,b)==1);
        pad=pad+1;
    y1=y((b-1)*(L+CP)+1:b*(L+CP),1);  % Loads L# I-Q data into vector
```

```
    y2=y1(CP+1:L+CP,1);                % Removes CP
    YR=fft(y2,L);       % L-point FFT converts time samples to I-Q data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Extracts user data from appropriate SC              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Y1(25:30,1)=YR(2:7,1);
Y1(31:43,1)=YR(9:21,1);
Y1(44:48,1)=YR(23:27,1);
Y1(1:5,1)=YR(39:43,1);
Y1(6:18,1)=YR(45:57,1);
Y1(19:24,1)=YR(59:64,1);
figure(2)                                    % NdataxNxm bits column wise
plot(Y1,'*');                                % Plot recieved I-Q data with
title(['Recieved I-Q Data'])                   %   noise
xlabel('Recieved In-Phase Data')
ylabel('Recieved Quadrature Data')
axis([-1.5 1.5 -1.5 1.5])
grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Generates 64QAM Demodulation Object                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if m==6
c=sqrt(42);
 hrec=modem.genqamdemod('Constellation', [ (-7-7j)/c,(7-7j)/c,...
    (-1-7j)/c, (1-7j)/c, (-5-7j)/c, (5-7j)/c, (-3-7j)/c, (3-7j)/c, ...
    (-7+7j)/c, (+7+7j)/c, (-1+7j)/c,(1+7j)/c, (-5+7j)/c, (5+7j)/c,...
    (-3+7j)/c, (3+7j)/c, (-7-1j)/c, (7-1j)/c, (-1-1j)/c,(1-1j)/c,...
    (-5-1j)/c, (5-1j)/c, (-3-1j)/c, (3-1j)/c  (-7+1j)/c,(7+1j)/c,...
    (-1+1j)/c,(1+1j)/c, (-5+1j)/c, (5+1j)/c, (-3+1j)/c, (3+1j)/c,...
    (-7-5j)/c, (+7-5j)/c, (-1-5j)/c  (1-5j)/c, (-5-5j)/c, (5-5j)/c,...
    (-3-5j)/c, (3-5j)/c, (-7+5j)/c, (7+5j)/c, (-1+5j)/c, (1+5j)/c,...
    (-5+5j)/c, (5+5j)/c, (-3+5j)/c, (3+5j)/c, (-7-3j)/c, (7-3j)/c,...
    (-1-3j)/c, (1-3j)/c, (-5-3j)/c, (5-3j)/c, (-3-3j)/c, (3-3j)/c,...
    (-7+3j)/c, (7+3j)/c, (-1+3j)/c, (1+3j)/c, (-5+3j)/c, (5+3j)/c,...
    (-3+3j)/c, (+3+3j)/c ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   Generates 16QAM Demodulation Object                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==4
c=sqrt(10);                              % Normalization Value of 16QAM Sym
    hrec=modem.genqamdemod('Constellation', [ (-3-3j)./c, ...
     (3-3j)./c, (-1-3j)./c, (1-3j)./c, (-3+3j)./c, (3+3j)./c,...
     (-1+3j)./c, (1+3j)./c, (-3-1j)./c, (3-1j)./c, (-1-1j)./c,...
     (1-1j)./c, (-3+1j)./c, (3+1j)./c,(-1+1j)./c, (1+1j)./c ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   Generates QPSK Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==2
c=sqrt(2);
     hrec=modem.genqamdemod('Constellation', [ (-1-1j)/c,  (1-1j)/c,
...
    (-1+1j)/c, (1+1j)/c  ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   Generates BPSK Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==1
     hrec=modem.genqamdemod('Constellation', [ -1, 1  ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');
    end                      % End of BPSK demod loop
    end                      % End of QPSK demod loop
    end                      % End of 16QAM demod loop
end                          % End of 64QAM demod loop
    end                      % End of else if (index(1,b)==1) loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Demodulates received data to baseband IQ data             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for b1=1:Ndata
    YRp=Y1(b1,1);
    Y1p(b1,1)=YRp;
    Ydemod=demodulate(hrec,YRp);        % Data back to n bit data and
    Yp(1,m*(b1-1)+1:m*b1)=Ydemod';
    end                                 % End of for b1=1:Ndata loop

ck(1,m*Ndata*(pad-1)+1:m*pad*Ndata)=Yp;
Y2((pad-1)*Ndata+1:pad*Ndata,1)=Y1p;
end                                    % End of if (index(1,b)==0) loop
end                                     % End of for b=1:N loop
bk=bk(1,1:length(ck));
err=bk(1,:)~=ck(1,:);                % Compares rec data string to trans
BER=max(cumsum(err))/(m*Ndata*N);
Pb=num2str(BER);
figure(3)                                % NdataxNxm bits column wise
plot(Y2,'*');                          % Plot recieved I-Q data with noise
title(['Recieved I-Q Data with BER of ',Pb,''])
xlabel('Recieved In-Phase Data')
ylabel('Recieved Quadrature Data')
axis([-1.5 1.5 -1.5 1.5])
grid on
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        Baseband Data Setup and Modulator                    %
%                           Written by Steven Schnur                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x]=WiMax_BasebandMod(L,N,m,Ndata,bk,CP,G,Frame)
if nargin~=8
    error('Wrong number of arguments')
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Generates 64QAM Modulation Object                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if m==6
c=sqrt(42);                                 % Normalization Value of 64QAM Sym
object = modem.genqammod('Constellation', [ (3+3j)/c,(3+j)/c,...
    (3+5j)/c, (3+7j)/c, (3-3j)/c, (3-j)/c, (3-5j)/c, (3-7j)/c, ...
    (1+3j)/c, (1+j)/c, (1+5j)/c,(1+7j)/c,(1-3j)/c,(1-j)/c,(1-5j)/c, ...
    (1-7j)/c, (5+3j)/c, (5+j)/c,(5+5j)/c,(5+7j)/c,(5-3j)/c,(5-j)/c, ...
    (5-5j)/c, (5-7j)/c (7+3j)/c,(7+j)/c,(7+5j)/c,(7+7j)/c,(7-3j)/c, ...
    (7-j)/c, (7-5j)/c,(7-7j)/c,(-3+3j)/c,(-3+j)/c,(-3+5j)/c  ...
    (-3+7j)/c, (-3-3j)/c, (-3-j)/c, (-3-5j)/c, (-3-7j)/c,(-1+3j)/c, ...
    (-1+j)/c, (-1+5j)/c, (-1+7j)/c, (-1-3j)/c, (-1-j)/c, (-1-5j)/c, ...
    (-1-7j)/c, (-5+3j)/c, (-5+j)/c, (-5+5j)/c, (-5+7j)/c,(-5-3j)/c, ...
    (-5-j)/c, (-5-5j)/c, (-5-7j)/c, (-7+3j)/c, (-7+j)/c, (-7+5j)/c, ...
    (-7+7j)/c, (-7-3j)/c, (-7-j)/c, (-7-5j)/c, (-7-7j)/c ], ...
    'InputType', 'Bit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Generates 16QAM Modulation Object                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==4
c=sqrt(10)                                  % Normalization Value of 16QAM Sym
object = modem.genqammod('Constellation', [ (1+j)./c, ...
     (1+3j)./c, (1-j)./c, (1-3j)./c, (3+j)./c, (3+3j)./c, (3-j)./c, ...
     (3-3j)./c, (-1+j)./c, (-1+3j)./c, (-1-j)./c, (-1-3j)./c,...
     (-3+j)./c, (-3+3j)./c,(-3-j)./c, (-3-3j)./c ], ...
    'InputType', 'Bit');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Generates QPSK Modulation Object                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==2
c=sqrt(2)
object = modem.genqammod('Constellation', [ (1+j)/c,  (-1+j)/c, ...
    (-1-j)/c, (1-j)/c ], ...
    'InputType', 'Bit');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Generates BPSK Modulation Object                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==1
object = modem.genqammod('Constellation', [ 1, -1 ], ...
        'InputType', 'Bit');
    end                       % End of BPSK mod loop
    end                       % End of QPSK mod loop
    end                       % End of 16QAM mod loop
end                           % End of 64QAM mod loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                Baseband Modulation                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

temp=bk;                                              % Dummy Vectors to load
pad=0;                                                %   preamble
index=mod([-1:N],Frame) & mod([0:N+1],Frame);  %Creats logic 4 preamble
                                                      %   insertion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  Outer loop to form overall tx data vector              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:N                                  % Loop to generate OFDM Symbols

     if ((index(1,k)==0) & (index(1,k+1)==0)); % Logic operation 2load
        [X]=preamble_1_64;                    %   64 x4 preamble symbol

        else if (index(1,k)==0)& (index(1,k+1)==1);% Log op 2 load
        [X]=preamble_2_128;                        %   128x2 preamb sym
        bkp=wextend('addcol','zpd',temp,m*2*Ndata,'r'); % Makes room in
        bkp=circshift(bkp,[0,m*2*Ndata]);           %   data vector for
        temp=bkp;                                   %   preamble

    else if index(1,k)==1;                       % Loads data if preamble
    x1=0;                                         %   not met
    a=bkp(1,Ndata*m*(k-1)+1:m*Ndata*k);     % Loads one symbol worth of
    xt=0;                                       %   data in a storage vec

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Inner loop to form OFDM symbol                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for w=1:Ndata                   % Loop to load m-by-Ndata fft vector
    aa=a(1,(m)*(w-1)+1:m*w);    % Grabs one m-symbol worth of data
    mk=aa(1,1:m);               % Loading of storage vect with 1 m-sym
    Xp=modulate(object,mk');  % Modulates m sized sym with obj: IQ data
    X1(w,1)=Xp;                 % Stacks IQ data in a column vector
end                             % End of w=1:Ndata loop
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      Pilot Subcarrier Sequence Generator                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PilotSeqGen =seqgen.pn('GenPoly', ...
    [1 0 0 0 0 0 0 0 0 1 0 1],...
    'InitialStates',[1 1 1 1 1 1 1 1 1 1 1],...
    'NumBitsOut',1);
set(PilotSeqGen, 'NumBitsOut', N);
wk=generate(PilotSeqGen);              % Creates PN Seq for Pilot SubCar
for k1=1:N;                              % Symbols as WIMAX Standard
if wk(k1,1)==0;                        % Logic loop to create compliment
    wkcomp(k1,1)=1;                    % of PN Seq for Negitive Pilot SC
else if wk(k1,1)==1;
        wkcomp(k1,1)=0;
    end
end
Negpilot(k1,1)=1-2*wk(k1,1);            % CreatesWIMAX Standard Pilot SC
Pospilot(k1,1)=1-2*wkcomp(k1,1);         % for pos and neg Pilot SC
end
NPSCsym=Negpilot(k,1);         % Loads Pilot data sym by sym
PPSCsym=Pospilot(k,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Load Vector X with Guard SC=0, DC=0, Pilot SC and Data in      %
%                            appropriate SC locations                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X(1,1)=0;                                % DC Null
X(2:12,1)=X1(97:107,1);                  % Data
X(13,1)=G*PPSCsym;                       % Positive Pilot SC
X(14:37,1)=X1(108:131,1);                % Data
X(38,1)=G*PPSCsym;                       % Positive Pilot SC
X(39:62,1)=X1(132:155,1);                % Data
X(63,1)=G*PPSCsym;                       % Positive Pilot SC
X(64:87,1)=X1(156:179,1);                % Data
X(88,1)=G*PPSCsym;                       % Positive Pilot SC
X(89:101,1)=X1(180:192,1);               % Data
X(102:128,1)=0;                          % Lower Guard Band
X(129:156,1)=0;                          % Upper Guard Band
X(157:168,1)=X1(1:12,1);                 % Data
X(169,1)=G*NPSCsym;                      % Negitive Pilot SC
X(170:193,1)=X1(13:36,1);                % Data
X(194,1)=G*NPSCsym;                      % Negitive Pilot SC
X(195:218,1)=X1(37:60,1);                % Data
X(219,1)=G*NPSCsym;                      % Negitive Pilot SC
X(220:243,1)=X1(61:84,1);                % Data
X(244,1)=G*NPSCsym;                      % Negitive Pilot SC
X(245:256,1)=X1(85:96,1);                % Data
Xplot((k-1)*L+1:k*L,1)=X;                % Dummy Vector for plotting
        end              % End of else if index(1,k)==1 loop
        end         % End if ((index(1,k)==0) & (index(1,k+1)==1)) loop
     end            % End if ((index(1,k)==0) & (index(1,k+1)==0)) loop
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           IFFT Baseband Modulated data to generate time data        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xt=ifft(X,L);       % L size IFFT to convert I-Q data to time samples
g(1:CP,1)=xt((L-CP+1):L,1); % Loads CP worth of Data from end of sym
x1(1:CP,1)=g;               % Loads CP to front of vector
x1(CP+1:L+CP,1)=xt;         % Loads rest of OFDM Symbol into vector
x(((k-1)*(L+CP)+1):k*(L+CP),1)=x1; % Fills transmit vector w/CP app
end                         % End for k=1:N loop
figure(1)
plot(X,'*');   % Plot Tx Constellation%title('Transmitted I-Q Data')
xlabel('Transmitter In Phase Data')
ylabel('Transmitted Quadrature Data')
axis([-1.5 1.5 -1.5 1.5]);
grid on


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           WiMax Preamble Construction: First Symbol                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X]=preamble_1_64;


pall=[ 1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 -
1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 -
1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1
+ 1i,1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,1 + 1i,1 +
1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,-1 - 1i,1
+ 1i,1 + 1i,1 + 1i,-1 + 1i,1 - 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,-1 -
1i,-1 - 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,-1
+ 1i,1 - 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,0 + 0i,-1 - 1i,1 + 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,1 +
1i,1 + 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,-1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,-
1 + 1i,1 - 1i,1 + 1i,1 + 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 +
1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,-1 + 1i,-1 + 1i,1 + 1i,-1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,-
1 - 1i,1 + 1i,-1 - 1i,-1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,-1 -
1i,1 + 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,-
1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,-1 + 1i,-1
+ 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i];

n=length(pall);
a=0;
ind=mod([0:n],4);
for k=1:n
if ind(k)==0
    a=a+1;
    pinsert51(a,1)=sqrt(2)*conj(pall(k,1));
end
end
```

```
p64=[pinsert51(1:25,1)',pinsert51(27:51,1)'];
%preamble4_64=[zeros(1,102:128),p64,p64,p64,p64];
%X=preamble4_64';
X(1,1)=0;
X(2:51,1)=p64;
X(52:101,1)=p64;
X(102:128,1)=0;
X(129:156,1)=0;
X(157:206,1)=p64;
X(207:256,1)=p64;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           WiMax Preamble Construction: First Symbol                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X]=preamble_2_128;
clear
clc
pall=[1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 -
1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 -
1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1
+ 1i,1 - 1i,1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,1 + 1i,1 +
1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,-1 - 1i,1
+ 1i,1 + 1i,1 + 1i,-1 + 1i,1 - 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,-1 -
1i,-1 - 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,1 - 1i,1 - 1i,-1
+ 1i,1 - 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,0 + 0i,-1 - 1i,1 + 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,1 +
1i,1 + 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,-1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,-
1 + 1i,1 - 1i,1 + 1i,1 + 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 +
1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,-1 + 1i,-1 + 1i,1 + 1i,-1 - 1i,1 - 1i,1 - 1i,1 + 1i,-1 - 1i,-1 - 1i,-
1 - 1i,1 + 1i,-1 - 1i,-1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,-1 -
1i,1 + 1i,-1 + 1i,-1 + 1i,-1 - 1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i,1 - 1i,-1 + 1i,-
1 + 1i,-1 + 1i,-1 + 1i,1 - 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,-1 - 1i,-1 - 1i,1 - 1i,-1 + 1i,-1 + 1i,-1
+ 1i,1 - 1i,-1 + 1i,1 + 1i,1 + 1i,1 + 1i,-1 - 1i,-1 - 1i,-1 - 1i,-1 - 1i,1 + 1i,1 - 1i,1 - 1i];

n=length(pall);
a=0;
ind=mod([0:n],2);
for k=1:n
if ind(k)==0
    a=a+1;
    pinsert101(a,1)=sqrt(2)*conj(pall(k,1));
end
end

p128=[pinsert101(1:50,1)', pinsert101(52:101,1)'];
%preamble2_128=[p128,p128];
%X=preamble2_128';
X(1,1)=0;
X(2:101,1)=p128;
```

```
X(102:128,1)=0;
X(129:156,1)=0;
X(157:256,1)=p128;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                WiMax Test Auto Fam (After [11])                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Sxp,fo,Np,centrow,centcol]=WiMax_autofamtest(famin,fs,df,...
    dalpha,ss,index,CP1,shift);
if nargin~=8
    error('Wrong number of arguments.')
end

for ww=3:index*2
    xin(1,(ss-CP1)*(ww-1)+1:(ss-CP1)*ww)=famin(1,ss*(ww-
1)+CP1+1:ss*ww);
end
%define parameters
Np=pow2(nextpow2(fs/df));
L=Np/4;
P=pow2(nextpow2(fs/dalpha/L));
N=P*L;
Sxp=zeros(Np+1,2*N+1);
for w=1:1:floor(index/2)
   x=xin(1,(shift)*(w-1)+1:(shift)*w);

%input channalization
if length(x)<N
    x(N)=0;
elseif length(x)>N
    x=x(1:N);
end
NN=(P-1)*L+Np;
xx=x;
xx(NN)=0;
xx=xx(:);
X=zeros(Np,P);
for k=0:P-1
    X(:,k+1)=xx(k*L+1:k*L+Np);
end

%windowing

a=hamming(Np);
XW=diag(a)*X;

%first FFT

XF1=fft(XW);
XF1=fftshift(XF1);
XF=[XF1(:,P/2+1:P) XF1(:,1:P/2)];
```

```
%downconvertions
E=zeros(Np,P);
for k=-Np/2:Np/2-1
    for k0=0:P-1
        E(k+Np/2+1,k0+1)=exp(-i*2*pi*k*k0*L/Np);
    end
end
XD=XF1.*E;
XD=conj(XD');

%multiplication
XM=zeros(P,Np^2);
for k=1:Np
    for c=1:Np
        XM(:,(k-1)*Np+c)=(XD(:,k).*conj(XD(:,c)));
    end
end

%second FFT
XF2=fft(XM);
XF2=fftshift(XF2);
XF2=[XF2(:,Np^2/2+1:Np^2) XF2(:,1:Np^2/2)];
XF2=XF2(P/4:3*P/4,:);
M=abs(XF2);
alphao=-fs:fs/N:fs;
fo=-fs/2:fs/Np:fs/2;
Sx=zeros(Np+1,2*N+1);
for k1=1:P/2+1
    for k2=1:Np^2
        if rem(k2,Np)==0
            c=Np/2-1;
        else
            c=rem(k2,Np)-Np/2-1;
        end
        k=ceil(k2/Np)-Np/2-1;
        p=k1-P/4-1;
        alpha=(k-c)/Np+(p-1)/L/P;
        f=(k+c)/2/Np;
        if alpha<-1 | alpha>1
            k2=k2+1;
        elseif f<-.5 | f>.5
            k2=k2+1;
        else
            kk=1+Np*(f+.5);
            ll=1+N*(alpha+1);
            Sx(round(kk),round(ll))=M(k1,k2);
        end
    end
end
Sxp=Sxp+Sx;
end

Sxp=Sxp./max(max(Sxp));
centrow=floor((Np+1)/2)+2;
centcol=round(1+N*(alpha/fs+1));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 WiMax Auto Fam (After [11])                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function
[Sxp,fo,Np,centrow,centcol]=WiMax_autofam(famin,fs,df,dalpha,...
    ss,index,v,shift,CP2);
if nargin~=9
    error('Wrong number of arguments.')
end
for ww=3:index*2
    xin(1,(ss-CP2)*(ww-1)+1:(ss-CP2)*ww)=famin(1,ss*(ww-
1)+CP2+1:ss*ww);
end
%define parameters
Np=pow2(nextpow2(fs/df));
L=Np/4;
P=pow2(nextpow2(fs/dalpha/L));
N=P*L;
Sxp=zeros(Np+1,2*N+1);
for w=1:1:floor(index/2)
   x=xin(1,(shift)*(w-1)+1:(shift)*w);

   %input channalization
if length(x)<N
    x(N)=0;
elseif length(x)>N
    x=x(1:N);
end
NN=(P-1)*L+Np;
xx=x;
xx(NN)=0;
xx=xx(:);
X=zeros(Np,P);
for k=0:P-1
    X(:,k+1)=xx(k*L+1:k*L+Np);
end

%windowing

a=hamming(Np);
XW=diag(a)*X;

%first FFT

XF1=fft(XW);
XF1=fftshift(XF1);
XF=[XF1(:,P/2+1:P) XF1(:,1:P/2)];

%downconvertions
E=zeros(Np,P);
for k=-Np/2:Np/2-1
    for k0=0:P-1
        E(k+Np/2+1,k0+1)=exp(-i*2*pi*k*k0*L/Np);
```

```
    end
end
XD=XF1.*E;
XD=conj(XD');

%multiplication
XM=zeros(P,Np^2);
for k=1:Np
    for c=1:Np
        XM(:,(k-1)*Np+c)=(XD(:,k).*conj(XD(:,c)));
    end
end

%second FFT
XF2=fft(XM);
XF2=fftshift(XF2);
XF2=[XF2(:,Np^2/2+1:Np^2) XF2(:,1:Np^2/2)];
XF2=XF2(P/4:3*P/4,:);
M=abs(XF2);
alphao=-fs:fs/N:fs;
fo=-fs/2:fs/Np:fs/2;
Sx=zeros(Np+1,2*N+1);
for k1=1:P/2+1
    for k2=1:Np^2
        if rem(k2,Np)==0
            c=Np/2-1;
        else
            c=rem(k2,Np)-Np/2-1;
        end
        k=ceil(k2/Np)-Np/2-1;
        p=k1-P/4-1;
        alpha=(k-c)/Np+(p-1)/L/P;
        f=(k+c)/2/Np;
        if alpha<-1 | alpha>1
            k2=k2+1;
        elseif f<-.5 | f>.5
            k2=k2+1;
        else
            kk=1+Np*(f+.5);
            ll=1+N*(alpha+1);
            Sx(round(kk),round(ll))=M(k1,k2);
        end
    end
end
Sxp=Sxp+Sx;
end

Sxp=Sxp./max(max(Sxp));

%    Surface Plot
centrow=floor((Np+1)/2)+2;
centcol=round(1+N*(alpha/fs+1));
Alpha=num2str(centcol);
Fo=num2str(centrow);
```

```
figure(4)
surfl(alphao,fo,Sxp);
view(-37.5,60);
xlabel('alpha');
ylabel('f');
zlabel('Sx');
colormap jet
colorbar
axis([0 3.5*10^6 -2*10^6 2*10^6 0 1])

%Contour Plot
figure(5)
contour(alphao,fo,Sxp,v);
xlabel('alpha,fo,Sxp');
ylabel('f(Hz)');
colormap winter
colorbar

%Cross-Section Plot
figure(6)
subplot(2,1,1), plot(alphao,Sxp(floor((Np+1)/2)+2,:));
xlabel('alpha(Hz)');
ylabel(['Sxp(',Fo,')'])
subplot(2,1,2), plot(fo,Sxp(:,round(1+N*(alpha/fs+1))));
xlabel('f(Hz)');
ylabel(['Sxp(',Alpha,')']);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    Baseband Data Setup and Demodulator                      %
%                          Written by Steven Schnur                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ck]=WiMax_BasebandDemod(L,N,m,Ndata,CP,y,bk,Frame)
if nargin~=8
    error('Wrong number of arguments')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Remove Preamble                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

index=mod([-1:N],Frame) ... % Logic vector to determine when to remove
    & mod([0:N+1],Frame);   %    Preamble
pad=0;
for b=1:N                  % Loop to break data string into L-fft sized
    Yp=0;                   %   blocks
    mpk=0;
    y1=0;

if (index(1,b)==0);         % Loop skips demod code if preamble logic
                            %   satisfied
else if (index(1,b)==1);
        pad=pad+1;
```

```
    y1=y((b-1)*(L+CP)+1:b*(L+CP),1);  % Loads L# I-Q data into vector
    y2=y1(CP+1:L+CP,1);               % Removes CP
    YR=fft(y2,L);       % L-point FFT converts time samples to I-Q data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Extracts user data from appropriate SC                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Y1(97:107,1)=YR(2:12,1);
Y1(108:131,1)=YR(14:37,1);
Y1(132:155,1)=YR(39:62,1);
Y1(156:179,1)=YR(64:87,1);
Y1(180:192,1)=YR(89:101,1);
Y1(1:12,1)=YR(157:168,1);
Y1(13:36,1)=YR(170:193,1);
Y1(37:60,1)=YR(195:218,1);
Y1(61:84,1)=YR(220:243,1);
Y1(85:96,1)=YR(245:256,1);
figure(2)                                   % NdataxNxm bits column wise
plot(Y1,'*');                               % Plot recieved I-Q data with
title(['Recieved I-Q Data'])                %   noise
xlabel('Recieved In-Phase Data')
ylabel('Recieved Quadrature Data')
axis([-1.5 1.5 -1.5 1.5])
grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Generates 64QAM Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if m==6
c=sqrt(42);
 hrec=modem.genqamdemod('Constellation', [ (3+3j)/c,(3+j)/c,...
    (3+5j)/c, (3+7j)/c, (3-3j)/c, (3-j)/c, (3-5j)/c, (3-7j)/c, ...
    (1+3j)/c, (1+j)/c, (1+5j)/c,(1+7j)/c,(1-3j)/c,(1-j)/c,(1-5j)/c, ...
    (1-7j)/c, (5+3j)/c, (5+j)/c,(5+5j)/c,(5+7j)/c,(5-3j)/c,(5-j)/c, ...
    (5-5j)/c, (5-7j)/c (7+3j)/c,(7+j)/c,(7+5j)/c,(7+7j)/c,(7-3j)/c, ...
    (7-j)/c, (7-5j)/c, (7-7j)/c, (-3+3j)/c, (-3+j)/c, (-3+5j)/c  ...
    (-3+7j)/c, (-3-3j)/c, (-3-j)/c, (-3-5j)/c, (-3-7j)/c,(-1+3j)/c, ...
    (-1+j)/c, (-1+5j)/c, (-1+7j)/c, (-1-3j)/c, (-1-j)/c,(-1-5j)/c, ...
    (-1-7j)/c, (-5+3j)/c, (-5+j)/c, (-5+5j)/c, (-5+7j)/c,(-5-3j)/c, ...
    (-5-j)/c, (-5-5j)/c, (-5-7j)/c, (-7+3j)/c, (-7+j)/c, (-7+5j)/c, ...
    (-7+7j)/c, (-7-3j)/c, (-7-j)/c, (-7-5j)/c, (-7-7j)/c ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Generates 16QAM Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==4
    hrec=modem.genqamdemod('Constellation', [ (1+j)./sqrt(10), ...
    (1+3j)./sqrt(10), (1-j)./sqrt(10), (1-3j)./sqrt(10), ...
    (3+j)./sqrt(10),(3+3j)./sqrt(10),(3-j)./sqrt(10),(3-3j)./sqrt(10),
...
```

```
    (-1+j)./sqrt(10), (-1+3j)./sqrt(10),  (-1-j)./sqrt(10), ...
    (-1-3j)./sqrt(10), (-3+j)./sqrt(10), (-3+3j)./sqrt(10),...
    (-3-j)./sqrt(10), (-3-3j)./sqrt(10 ) ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  Generates QPSK Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==2
     hrec=modem.genqamdemod('Constellation', [ (1+j)/sqrt(2),  ...
              (-1+j)/sqrt(2), (-1-j)/sqrt(2), (1-j)/sqrt(2)  ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  Generates BPSK Demodulation Object                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

else if m==1
     hrec=modem.genqamdemod('Constellation', [ 1, -1  ], ...
    'OutputType', 'Bit', 'DecisionType', 'hard decision');
    end                      % End of BPSK demod loop
    end                      % End of QPSK demod loop
    end                      % End of 16QAM demod loop
end                          % End of 64QAM demod loop
    end                      % End of else if (index(1,b)==1) loop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              Demodulates received data to baseband IQ data             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for b1=1:Ndata
    YRp=Y1(b1,1);
    Y1p(b1,1)=YRp;
    Ydemod=demodulate(hrec,YRp);        % Data back to n bit data and
    Yp(1,m*(b1-1)+1:m*b1)=Ydemod';
    end                                 % End of for b1=1:Ndata loop

ck(1,m*Ndata*(pad-1)+1:m*pad*Ndata)=Yp;
Y2((pad-1)*Ndata+1:pad*Ndata,1)=Y1p;
end                                    % End of if (index(1,b)==0) loop
end                                     % End of for b=1:N loop
bk=bk(1,1:length(ck));
err=bk(1,:)~=ck(1,:);                % Compares rec data string to trans
BER=max(cumsum(err))/(m*Ndata*N);
Pb=num2str(BER);
figure(3)                               % NdataxNxm bits column wise
plot(Y2,'*');                         % Plot recieved I-Q data with noise
title(['Recieved I-Q Data with BER of ',Pb,''])
xlabel('Recieved In-Phase Data')
ylabel('Recieved Quadrature Data')
axis([-1.5 1.5 -1.5 1.5])
grid on
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] S. H. Sohn, N. Han, J. M. Kim, and J. W. Kim, "OFDM signal sensing method based on cyclostationary detection," Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007, pp. 63–68, 1-3 August 2007.

[2] W. A. Gardner, "Cyclostationarity in communications and signal processing," IEEE Press, Piscataway, NJ, 1994.

[3] K. Maeda, A. Benjebbour, T. Asai, T. Furuno, and T. Ohya,"Cyclostationarity-Inducing Transmission Methods for Recognition among OFDM-Based Systems," EURASIP Journal on Wireless Communications and Networking, Vol. 2008.

[4] P. D. Sutton, K.E. Nolan, and L. E. Doyle, "Cyclostationary Signatures in Practical Cognitive Radio Applications," *IEEE Journal on Selected Areas in Communications*, VOL. 26, NO. 1, pp. 13–24, 2008.

[5] J. G. Andrews, A. Ghosh, and R. Muhamed, "Fundamentals of WiMAX Understanding Broadband Wireless Networking," Prentice Hall, Upper Saddle River, New Jersey 07458, 2007.

[6] M. Tummala, Lecture notes for EC-4940 "Mobile Ad-Hoc Networks," Naval Postgraduate School, Monterey, CA Spring 2009.

[7] Institute of Electrical and Electronics Engineers, 802.16, Air Interface for Fixed Broadband Wireless Access Systems, 1 October 2004. http://ieeexplore.ieee.org, accessed 5 June 2009.

[8] Institute of Electrical and Electronics Engineers, 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. http://ieeexplore.ieee.org, accessed 5 June 2009.

[9] C. W. Therrien, and M. Tummala, "Probability for Electrical and Computer Engineers," CRC Press LLC, Boca Raton, Florida, 2004.

[10] R. S. Roberts, W. A. Brown, and H. H. Loomis, "Computationally Efficient Algorithms for Cyclic Spectral Analysis," *IEEE Signal Processing Magazine*, pp. 38–49, April 1991.

[11] E. L. da Costa, "Detection and identification of cyclostationary signals," Master's thesis, Naval Postgraduate School, Monterey, CA, 1996.

[12] K. C. Howland, “Signal detection and frame synchronization of multiple wireless networking waveforms,” Master’s thesis, Naval Postgraduate School, Monterey, CA, 2007.

[13] B. Sklar, “ Digital communications fundamentals and applications,” Prentice Hall, Upper Saddle River, New Jersey 07458, 2001.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education
   MCCDC, Code C46
   Quantico, Virginia

5. Dierector, Marine Corps Research Center
   MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

7. Kirkpatrick, Thomas
   SPAWAR SSC ATLANTIC
   Charleston, South Carolina

8. Niermann, Michael
   SPAWAR SSC ATLANTIC
   Charleston, South Carolina

9. Riggins, Michael
   CDR USSOCOM HQ
   McDill AFB, Florida

10. John G. Kato
    Naval Information Operations Command Suitland
    Suitland, Maryland